

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматизації та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Програмне забезпечення
інформаційно-комунікаційних систем»
спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Мобільний застосунок для тренування
швидкості набору тексту»**

Виконав:

студент IV курсу, групи ІТ-61,
Авраменко Артем Сергійович

Керівник:

старший викладач каф. АУТС,
Март Богдан Анатолійович

Рецензент:

доцент каф. ОТ, к.т.н.,
Волокита Артем Миколайович

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Авраменку Артему Сергійовичу

1. Тема проєкту «Мобільний застосунок для тренування швидкості набору тексту», керівник проєкту Март Богдан Анатолійович, старший викладач, затверджені наказом по університету від «07» травня 2020 р. №1081-с.

2. Термін подання студентом проєкту 9 червня 2020 року

3. Вихідні дані до проєкту: мобільний телефон, операційна система Android, перевірка та оцінка друку користувача, збереження результатів.

4. Зміст пояснювальної записки: Вступ, Дослідження предметної області, Визначення вимог до проєкту, Технології та рішення використані при розробці, Реалізація додатку, Тестування, Вимоги до технічного засобу та інструкція користувача.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Схема ієрархія об'єктів в сцен, Діаграма послідовності дій користувача, Блок-схема анімацій, Діаграма класів, Блок-схема рівню.

6. Дата видачі завдання 15 лютого 2020 року.

Календарний план

№ з\п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Отримання завдання	15.02.20	
2	Збір інформації та аналіз існуючих рішень	5.03.20	
3	Розробка прототипу додатку	3.04.20	
4	Розробка графічного складової	24.04.20	
5	Тестування створеної системи	8.05.20	
6	Випуск додатку в Play Store	22.05.20	
7	Оформлення дипломного проєкту	8.06.20	

Студент

Артем АВРАМЕНКО

Керівник

Богдан МАРТ

АНОТАЦІЯ

Авраменко А.С. Мобільний застосунок для тренування швидкості набору тексту. КПІ ім. Ігоря Сікорського Київ, 2020.

Проект містить 60 ст. тексту, 21 рисуноків, 6 таблиць, посилання на 17 літературних джерел та 4 додатки.

Ключові слова: мобільний додаток, тренажер, набір тесту, ігровий рушій, телефон.

Об'єктом розробки є програмне забезпечення для телефонів, яке є тренажером набору тексту.

Мета розробки – полегшення отримання навичку швидкого друку на телефоні.

У дипломному проєкті буде розроблено програмне забезпечення для телефонів. Воно міститиме серію вправ, після проходження яких, користувач зможе покращити навички швидкого друку. Буде проведено ретельний аналіз для вибору форми, в якій подаватимуться вправи та як вправи будуть побудовані. Буде розроблений зручний, швидкий та приємний інтерфейс. Також, прицілиться багато уваги вибору ігрового двигуна та мови програмування для розробки мобільного додатку.

Отримані результати можуть бути корисними користувачами мобільних пристроїв для покращення навичку набору тексту. Додаток буде опубліковано в Google Play Store.

SUMMARY

Avramenko A.S. Mobile application for training speed typing. Igor Sikorsky Kyiv Polytechnic Institute, 2020.

The project contains 60 p. Text, 21 figures, 6 tables, references to 17 literature sources and 4 applications.

Keywords: mobile application, simulator, test set, game engine, phone.

The object of development is software for phones, which is a typing simulator.

The purpose of the development is to facilitate the acquisition of the skill of fast typing on the phone.

The diploma project will develop software for phones. It will contain a series of exercises, after which the user will be able to improve their typing skills. A thorough analysis will be conducted to select the form in which the exercises will be presented and how the exercises will be constructed. A user-friendly, fast and pleasant interface will be developed. Also, a lot of attention will be paid to the choice of game engine and programming language for the development of a mobile application.

The results can be useful for mobile users to improve typing skills. The application will be published in the Google Play Store.

№ рядка	Формат	Позначення	Найменування	Кіл. листів	№ прим.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	A4	IT61.010БАК.004 ПЗ	Мобільний застосунок для			
6			тренування швидкості			
7			набору тексту.			
8			Пояснювальна записка	60		
9	A3	IT61.010БАК.004 Д2	Мобільний застосунок для			
10			тренування швидкості			
11			набору тексту.			
12			Діаграма класів	1		
13	A3	IT61.010БАК.004 Д3	Мобільний застосунок для			
14			тренування швидкості			
15			набору тексту.			
16			Схема архітектури			
17			ігрового рушія	1		
18	A3	IT61.010БАК.004 Д4	Мобільний застосунок для			
19			тренування швидкості			
20			набору тексту.			
21			Блок-схема рівню	1		
22	A3	IT61.010БАК.004 Д1	Мобільний застосунок для			
23			тренування швидкості			
24			набору тексту.			
25			Блок-схема анімацій гравця	1		
26						
27						
28						
29						
Змн.	Лист	№ докум.	Підпис	Дата	IT61.010БАК.004 ТП	
Розроб.	Авраменко А.С.				Мобільний застосунок для тренування швидкості набору тексту.	
Перевір.	Март Б.А.					
					КПІ ім. Ігоря Сікорського кафедра АУТС гр. IT-61	
Н. Контр.						
Затверд.						
					Відомість технічного проекту	

**Пояснювальна записка
до дипломного проєкту
на тему: «Мобільний застосунок для тренування
швидкості набору тексту»**

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП.....	6
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Загальний огляд.....	8
1.2 Аналіз існуючих рішень	8
1.2.1 Сайт typing.com	8
1.2.2 Сайт play.typeracer.com	9
1.3 Мобільна гра “Typing zombie”.....	11
1.4 Мобільна гра “Typing master”.....	12
Висновки до розділу 1	14
2 ВИЗНАЧЕННЯ ВИМОГ ДО ПРОЄКТУ	17
2.1 Функціональні можливості	17
2.2 Функціонал для розробників	18
2.3 Користувацький інтерфейс	19
Висновки до розділу 2	20
3 ТЕХНОЛОГІЇ ТА РІШЕННЯ ВИКОРИСТАНІ ПРИ РОЗРОБЦІ.....	21
3.1 Ігровий рушій на Unity.....	21
3.1.1 Розробка на Unity	22
3.1.1.1 Сцени, ігрові об’єкти та компоненти	22
3.1.1.2 Prefabs	23
3.1.1.3 Створення екземплярів(instancing).....	23
3.1.2 Архітектура ігор в Unity	24
3.1.3 Переваги Unity над іншими рушіями.....	25
3.2 Мова програмування C#	26

					ІТ61.010БАК.004 ПЗ				
Зм.	Арк.	№ докум.	Підпис	Дата	Мобільний застосунок для тренування швидкості набору тексту. Пояснювальна записка	Літера		Аркуш	Аркушів
Розроб.		Авраменко А.С.				Т		2	60
Перевір.		Март Б.А.				КПІ ім. Ігоря Сікорського кафедра АУТС гр. ІТ-61			
Т.контр.									
Затвер.									

3.2.2 C# та Unity	27
3.3 Open Graphics Library	28
3.4 TexturePacker	30
Висновки до розділу 3	31
4 РЕАЛІЗАЦІЯ ДОДАТКУ	32
4.1 Реалізація рівня	32
4.2 Реалізація Head-Up Display	35
4.3 Анімації.....	36
4.3.1 Система Animation в Unity	36
4.3.2 Анімація пересування по рівню	39
4.4 Графіка.....	40
4.5 Інструменти для розробників	42
4.5.1 Налаштування рівня	42
4.5.2 Система створення перешкод	43
4.6 Інтерфейс	44
4.6.1 Проєктування інтерфейсу	44
4.6.2 Реалізація інтерфейсу.....	45
4.7 Збереження прогресу користувача	47
Висновок до розділу 4	48
5 ТЕСТУВАННЯ.....	49
5.1 Тести анімацій персонажу та перешкоди	50
Висновок до розділу 5	53
6 ВИМОГИ ДО ТЕХНІЧНОГО ЗАСОБУ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА...	54
6.1 Мінімальна конфігурація	54
6.2 Інструкція користувача	54
6.2.1 Установка програми	54
Висновок до розділу 6	56

ВИСНОВКИ57

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....59

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

XML – eXtensible Markup Language (розширювана мова розмітки)

OpenGL - Open Graphics Library (відкрита бібліотека графіки)

HUD - Head-Up Display

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		5

ВСТУП

Починаючи з дати винаходу друкарської машини люди намагалися покращити свої навички друкування. На той час це була їх робота, а друкування - їх професійним навиком. З винаходом комп'ютера, все у більшій кількості людей виникала потреба в розвитку цього навика не тільки для професійної діяльності, а і для спілкування. Зараз смартфон, той самий комп'ютер, проник в більшість сфер життя. І доказом цього є те, що на сьогоднішній день нараховують 4 мільярди смартфонів на землі. Не тільки збільшилась кількість смартфонів, люди стали більше користуватися ними. Смартфон вже для багатьох став персональним помічником, який вони ніколи не залишають. Тому існує велика потреба в навичку швидкого набору на смартфонах, адже клавіатура є одним з найголовніших інструментів смартфона.

Актуальність даної теми зумовлена тим, що зараз не існує рішення серед мобільних додатків, яке б не мало великої кількості недоліків. Серед найчастіших недоліків можна виділити:

- ігрові механіки, які заважають ігровому процесу;
- застарілий та незручний інтерфейс;
- відсутність різноманітності.

Та саме зараз, з виходом потужних ігрових рушіїв, з'явилась можливість створити додаток під мобільну платформу, який би не мав би всіх перерахованих недоліків.

Метою даного дипломного проєкту стала розробка тренажеру для друку, який мав би цікавий процес навчання в формі гри, також мав зручний і інтуїтивно зрозумілий інтерфейс та містив велику кількість різноманітних навчальних вправ.

На даний проєкт було винесено наступні завдання:

- дослідження предметної області, для визначення критеріїв за якими буде оцінюватися проєкт та аналогічні додатки. Огляд переваг та недоліків існуючих рішень;

					IT61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		6

- визначення вимог до проєкту, на основі проведеного дослідження існуючих рішень;
- огляд технологій;
- реалізація додатку. Розгляд рішень, які були прийняті при розробці додатку, опис реалізації програми та окремих її компонентів;
- тестування. Тестування додатку на різних девайсах.

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		7

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний огляд

В цьому розділі будуть розглядатися програми, які досягають успіху різними способами. Для ефективного навчання друку головними є два критерії:

- наскільки метод навчання цікавий для користувача;
- ефективність методу навчання.

Також важливими критеріями оцінки будуть:

- ефективність методу навчання;
- різноманіття вправ.

1.2 Аналіз існуючих рішень

Вже існує багато рішень для навчання швидкого набору тексту на комп'ютері та декілька телефонах. Для реалізації всіх поставлених задач, потрібно спочатку оглянути вже існуючі рішення. В кожному з них є свої переваги та недоліки. Аналіз цих особливостей важливий для розробки додатку. Серед аналогів будуть розглянуті сайт typing.com, сайт play.typeracer.com, гра [typing zombie](http://typingzombie.com).

1.2.1 Сайт typing.com

Першим буде розглянутий сайт typing.com. Він є одним за лідерів на ринку сайтів, як тренажерів для набору тексту. Приклад інтерфейсу зображено на рисунку 1.1.

Сайт зображений на рисунку 1.1 має простий та зрозумілий інтерфейс. Тут всі завдання поділені по складності на три групи, користувач вибирає один із них та тренується на задачах, які відповідають цьому рівню складності. Таким чином сайт може бути корисним для людей з різним навиком набору тексту.

					ІТ61.010БАК.004 ПЗ	Аркуш
						8
Зм.	Арк.	№ документа	Підпис	Дата		

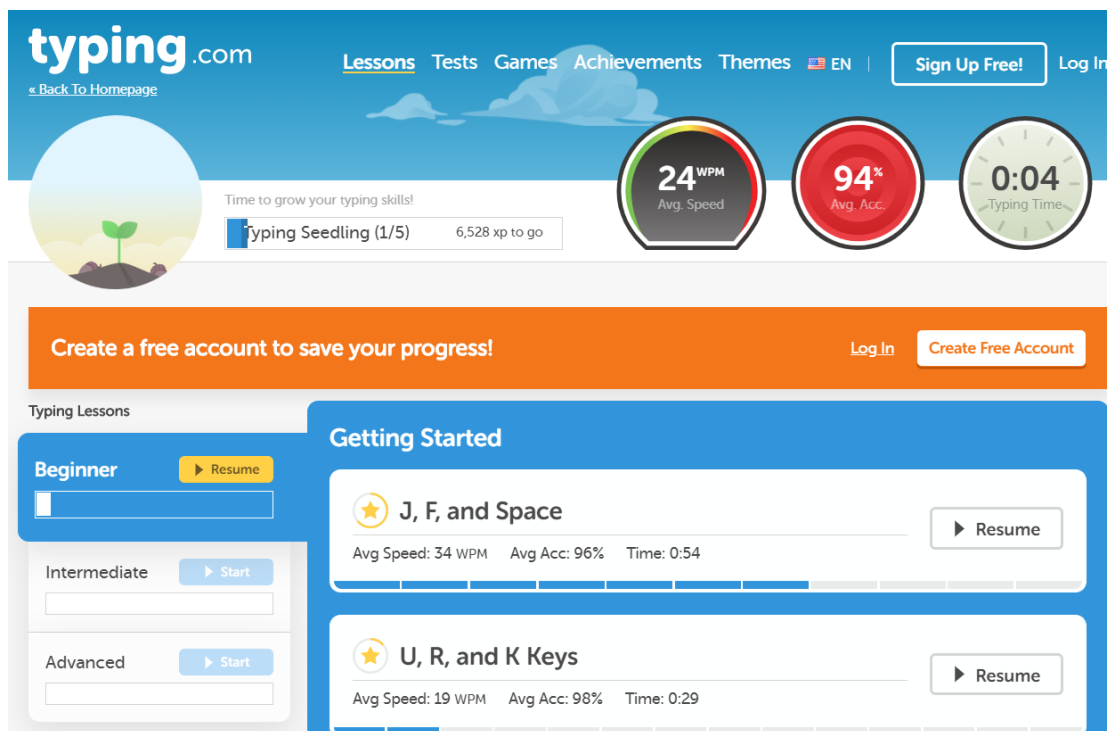


Рисунок 1.1 – Сайт typing.com

Також сайт володіє декількома механіками, які роблять проходження завдань цікавішими. На рисунку 1.1 в правому верхньому куту можна побачити: скільки часу ви вже тренуєтесь, середню швидкість та точність набору тесту. Ця статистика заохочує гравців покращувати свій результат та продовжувати тренуватися.

Також я хочу відмітити зручний та зрозумілий інтерфейс при проходженні завдань. При натиску будь якої клавіши інтерфейс зміниться та дасть тобі зрозуміти якими повинні бути твої наступні дії.

Сайт typing.com є безкоштовним, та фінансується за допомогою реклами, але має платну опцію для прибирання реклами. Цей варіант монетизації надає можливість набрати велику кількість користувачів.

1.2.2 Сайт play.typeracer.com

Другим буде розглянутий сайт play.typeracer.com. Цей сайт існує вже досить довго. Тренажер тут представлений як гра в браузері. Веб платформа та проста реалізація дозволила цьому сайту знайти свою аудиторію та стати популярним

					IT61.010БАК.004 ПЗ	Аркуш
						9
Зм.	Арк.	№ документа	Підпис	Дата		

платформою для змагань по швидкості набору тексту. Часто ведучі трансляцій люди заходять на даний сайт, щоб їх глядачі могли позмагатися з ними в наборі тесту. Сайт зображений на рисунку 1.2.

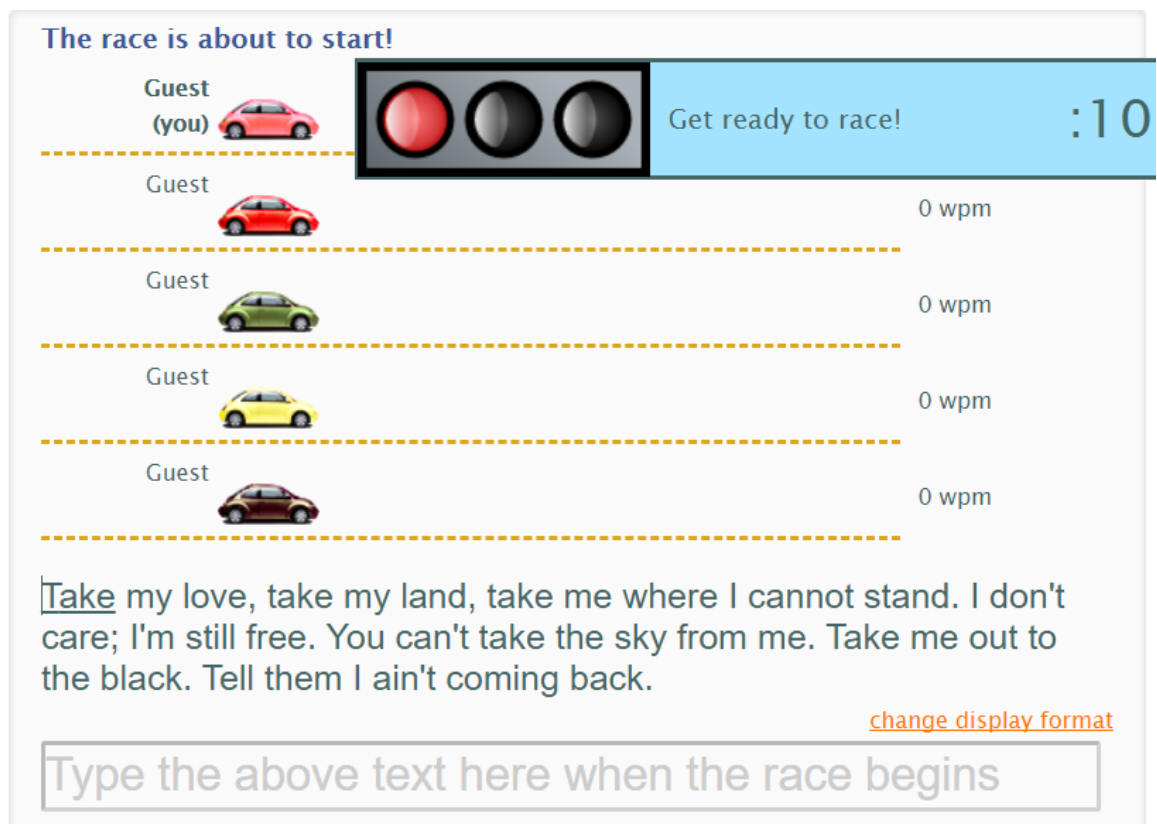


Рисунок 1.2 – сайт play.typeracer.com

Це популярний сайт, на якому користувачі можуть змагатися в швидкості набору тексту. Декілька гравців заходять на сайт в один той самий час та починають гру. Сайт їх поєднує в одному матчі та вони повинні на швидкість написати параграф тексту. Після завершення матчу гравцям присвоюються призові місця та зберігається результат з яким закінчили гру.

Саме аспект змагання з справжніми людьми перетворює монотонний набір тесту в щось цікаве та за рахунок цього режиму гри сайт набрав таку популярність.

Тому я можу виділити такі переваги:

- веб платформа;
- безкоштовна гра;
- змаганнях між гравцях;

					IT61.010БАК.004 ПЗ	Аркуш
						10
Зм.	Арк.	№ документа	Підпис	Дата		

– простий інтерфейс.

1.3 Мобільна гра “Typing zombie”

Наступною буде розглянута гра “Typing zombie”. Це приклад проекту розробленого для мобільної платформи. Також розробники представили тренажер для набору тексту в більш ігровій формі. Гра зображена на рисунку 1.3.



Рисунок 1.3 – Мобільна гра “Typing zombie”

Цей додаток не має володіє великою популярністю та його можна скоріше назвати невдачею. Але для нас важливо розглянути додатки, які не змогли реалізувати тренажер на мобільних телефонах, вияснити що призвело до невдачі та розібрати критичні помилки, які були допущені розробниками під час розробки даного додатку.

Перевагами цього додатку є те, що він випущений на мобільній платформі та була створена цікава ігрова механіка, яка примушує гравця друкувати все швидше з плином часу, що гейміфікує набір тексту.

Також до переваг даного додатку можна віднести його тему. Оформлення гри в битву проти зомбі додає інтересу до ігрового процесу та може привабливим

					ІТ61.010БАК.004 ПЗ	Аркуш
						11
Зм.	Арк.	№ документа	Підпис	Дата		

для молоді аудиторії.

Та тема цього додатку має свої недоліки. Зомбі більше імпонують молодій аудиторії та відштовхують всіх інших користувачів.

Найбільший недолік даної гри – це відсутність чіткої стратегії у розробників. Не зрозуміло, що з себе повинен представляти даний продукт, гру з керуванням клавіатурою або це тренажер для набору тексту з деякими ігровими механіками.

Також критичним недоліком, який був допущений в цьому додатку, є використання не вбудованої клавіатури. Це означає, що гравець не може перенести отриманий в даній грі навик швидкого друку та застосувати його в інших додатках.

Також можна помітити низка помилок в цьому додатку. Список недоліків даного мобільного додатку:

- не розбірливий інтерфейс;
- відсутність різноманіття в завданнях;
- тільки одним режим гри;
- відсутність рівнем складності;
- не використання базової клавіатури в телефоні;
- гра погано адаптована під горизонтальний режим;
- відсутність анімацій.

Це досить проста гра, але на її прикладі видно, які складнощі можуть виникнути при перенесенні ідей та ігрових механік, які були протестовані на комп'ютерах та веб платформі, на мобільні девайси.

1.4 Мобільна гра “Typing master”

Наступним буде розглядатися гра для мобільних телефонів “Typing master”. Це успішний додаток для телефонів, який має більше ніж сто тисяч завантажуваль. Додаток “Typing master”, зображений на рисунку 1.4.

					IT61.010БАК.004 ПЗ	Аркуш
						12
Зм.	Арк.	№ документа	Підпис	Дата		



Рисунок 1.4 – Мобільна гра “Typing master”

Цей додаток містить велику кількість режимів гри, але всі режими мають спільний елемент – клавіатура та набір слів. Тут тренування набору тексту не займає не головну задачу. Розробники більше сконцентрували зусиль на тому, щоб зробити гру веселою.

Ми можемо побачити схожі рішення до рішень прийнятих при розробці сайту typing.com та play.typeracer.com.

Список схожих рішень:

- слова з’являються поступово перед користувачем;
- кожна літера підсвічується червоним, зеленим або сірим кольором, що чітко демонструє гравцю помилки, скільки літер він набрав та на якій літері він зупинився;
- доданий таймер, який обмежує гравця в часі та змушує його швидше друкувати;
- має декілька режимів гри.

В даному додатку також були застосовані нові механіки. Гравець може допустити за рівень обмежену кількість помилок. Також користувач має доступ до обмеженої кількості здібностей, які різним чином можуть допомогти пройти

					IT61.010БАК.004 ПЗ	Аркуш
						13
Зм.	Арк.	№ документа	Підпис	Дата		

рівень.

Але в цьому популярному додатку були допущені помилки схожі на помилки допущені в грі “Typing zombie”.

Першою помилкою є використання своєї клавіатури. Це значно зменшує ефективність навчання та не дає змоги користувачу використовувати отримані навички швидкого друку.

Другою є додання великої кількості ігрових механік. Здібності, які допомагають гравцеві, вони виконують тільки розважальну задачу і завдають шкоди навчальному процесу.

Також важливим є вибір розробниками стратегії по монетизації додатку. Вони не тільки розміщують рекламу, а ще й пропонують користувачу купити здібності за гроші. Вибір такої стратегії спричинив введення механік, які шкодять навчальному процесу.

Всі помилки виходять з невірного фокусу розробників. В іграх клавіатура виступає як контролер грою, не як інструмент, якому потрібно навчитися. Це і призводить до більшості важливих помилок.

Висновки до розділу 1

Розглянуті системи мають зовсім різні цілі та застосовують різні стратегії. Проаналізувавши всі чотири вибрані системи, було сформовано таблицю 1.1, яка демонструє переваги одних систем над іншими за різними критеріями.

Таблиця 1.1 – Порівняльний аналіз існуючих рішень за критеріями

Критерій\Назва програми	typing.com	play.typeracer.com	Typing zombie	Typing master
Наскільки метод навчання цікавий для користувача	5	3	2	5

Ефективність методу навчання	3	5	2	3
Зрозумілість інтерфейсу	4	5	2	4
Різноманіття	2	4	1	3

Головною відмінністю між розглянутими системами є підхід набору тексту. typing.com та play.typeracer.com розглядають набір тексту більше як навик, який користувач захоче або покращити, або позмагатися з іншими користувачами.

З іншої сторони, “Typing zombie” та “Typing master” розглядають набір тексту та клавіатуру, як інтерфейс для керування грою. Навик набору тексту стоїть для них не на головному місці та вони хочуть зробити просто цікаву гру.

Обрана стратегія typing.com та play.typeracer.com набагато більше підходить для проєкту. Але “Typing zombie” та “Typing master” володіють рядом функція, які не заходили б навчальному процесу та зробили би додаток цікавішим.

Проаналізувавши всі чотири вибрані системи, було сформовано таблицю 1.2, яка демонструє реалізовані в системах функції.

Таблиця 1.2 – Порівняльний аналіз існуючих рішень за наявністю функціоналу

Функції\Назва програми	typing.com	play.typeracer.com	typing zombie	typing master
Декілька вправ	+	-	-	+
Змагання	-	+	-	-
Обмеження по часу	-	-	+	+

Статистика по точність набору	+	+	-	-
Статистика по швидкості набору	+	+	-	-
Додаткові здібності у гравця	-	-	-	+
Додаткові режими гри	+	-	-	+

В проєкті потрібно поєднати існуючі механіки в typing.com і play.type racer.com та застосувати ігрові механіки з “Typing zombie” та “Typing master”, які не заважають навчальному процесу. Також потрібно уникнути допущених помилок в розглянутих системах.

2 ВИЗНАЧЕННЯ ВИМОГ ДО ПРОЄКТУ

У попередньому розділі було проведено аналіз недоліків та переваг вже існуючих веб застосунків і мобільних ігор призначених для тренування навичку друкування.

З попереднього розділу видно, що представлені проєкти досить різні та привели їх до успіху різні рішення. Потрібно взяти приклад з найкращих рішень, вдало їх поєднати та вдало перенести на мобільні девайси.

На основі зробленого аналізу існуючих рішень було сформовані вимоги до майбутнього мобільного додатку.

2.1 Функціональні можливості

Список функцій, які повинні бути реалізовані в додатку:

- реалізувати вибір складності гравцем. Він повинен бути реалізований як прогресія гравця між рівнями. Та складність завдань повинна бути розрахована на користувачів з різними навиками друкування. Таким чином, користувач який вже має навички, матиме можливість пропустити перші рівні в грі;
- зберігати прогрес гравця. Кількість пройдених завдань та результат з яким гравець пройшов ці завдання повинні зберігатися на мобільному девайсі користувача;
- розробити елементи управління для вправ, які були б зручними в використанні та зрозумілими гравцю при першому ж використанні, не потребували ніякого навчання перед використанням;
- демонструвати точність набору. Точність набору є одним з результатів, який зберігається після набору рівнів та демонструється гравцю під час проходження завдання. Користувач повинен мати можливість повернутися та покращити свій результат;
- реалізувати прогрес між завданнями в додатку. Повинні відкриватися наступні рівні по мірі покращення гравцем навичку, ці рівні будуть все

					IT61.010БАК.004 ПЗ	Аркуш
						17
Зм.	Арк.	№ документа	Підпис	Дата		

складнішими та націлені на різні цілі. Таким чином у гравця буде прогресія під час використання додатку;

- реалізувати автоматичне налаштування складності рівня під час гри. Якщо гравець дуже швидко справляється з перешкодами, тоді рівень повинен трохи прискорюватись. Не повинно бути моменту коли гравець пройшов всі перешкоди та чекає на наступну перешкоду.

2.2 Функціонал для розробників

Для навчання потрібно створити велику кількість рівнів. Ці рівні повинні бути різноманітними та відрізнятися одне від одного. Тому виникає потреба в створенні функціоналу для розробників, який би дозволив швидко розробляти різноманітні рівні. Також розробка нових рівнів посприяла би в підтримці інтересу до проекту після його випуску:

- всі перешкоди, перед гравцем повинні створюватись динамічно та залежати від тексту, який залишилось гравцеві написати. Потрібно щоб була реалізована можливість налаштувати новий рівень, тільки змінивши текст та налаштування швидкості;

- додання нового рівня повинно бути зручним та легким. Для якісного навчального процесу потрібна велика кількість рівнів. Також для підтримання інтересу до додатку, будуть додаватись нові рівні. Це є важливою частиною даного додатку та повинен бути розроблені інструменти, які б прискорювали процес додання нового рівня;

- кожен рівень повинен зовнішньо відрізнятися від інших. До рівня додаватимуться тільки ті гравійні елементи, які мають декілька варіантів дизайну та можуть бути з легкістю змінені поміж рівнями;

- зміна зовнішнього вигляду рівня також повинно бути зручним. Таким чином повинні бути розроблені інструменти, які полегшують процес зміни зовнішнього вигляду рівня;

- всі графічні елементи повинні мати декілька аналогів виконаних в

					IT61.010БАК.004 ПЗ	Аркуш
						18
Зм.	Арк.	№ документа	Підпис	Дата		

такому ж стилі. Таким чином можливе використання одних елементів на різних рівнях або зміни зовнішнього виду рівня, замінивши всі графічні елементи на аналогічні.

2.3 Користувацький інтерфейс

Користувацький інтерфейс є важливою частиною розроблюваної системи. В даній грі будуть два інтерфейси. Перший відповідає за зміни на рівні. Другий за переходи між меню та інші елементи.

Якщо перед другим інтерфейсом стоїть задача тільки бути зручним та естетичним. То інтерфейс, який знаходиться в рівні повинен швидко та в зрозумілому вигляді показувати інформацію гравцю.

Наступний перелік вимог до проектування інтерфейсу гри:

- інтерфейс повинен давати чітку та зрозумілу інформацію користувачу під час виконання завдання. Гравцю потрібно відображати правильно, неправильно набрані літери, текст який ще потрібно набрати та на якій літері зупинився гравця;
- під час проходження рівня, повинні відображатися різні показники. Такі як: точність друку, процент наскільки пройдено завдання. Вони повинні залишатися на другому плані та не заважати гравцю;
- гра повинна мати меню для вибору ріней. Там потрібно відображати інформацію про рівні;
- всі зображення та механіки повинні бути приємними для широкої аудиторії та відповідати критеріям вікового рейтингу 6+;
- усі схожі елементи управління та графіки повинні мати однакові ефекти та виконані в одному стилі. Допускаються невеликі відмінності, щоб підкреслити важливі елементи інтерфейсу та графіки;
- інтерфейс повинен бути адаптований під всі можливі розміри мобільного телефону;
- інтерфейс не буде підтримувати горизонтальне положення телефону та

					IT61.010БАК.004 ПЗ	Аркуш
						19
Зм.	Арк.	№ документа	Підпис	Дата		

буде працювати тільки в вертикальному.

2.4 Графіка та анімації

Під графікою мається на увазі всі графічні елементи які не відповідають за інтерфейс. Графіка та анімації є важливою частиною будь-якої гри. Вони не тільки повинні бути естетичними а й доносити інформацію до гравця. Тому був сформований перелік вимог до проєктування інтерфейсу гри.

Перелік вимог:

- важливі елементи на рівні повинні бути анімовані;
- анімації повинні показувати змін стану гравця;
- анімація гравця та перешкод повинна бути швидкою та динамічною.

Повинна відпрацьовувати одразу після зміни стану;

- всі зображення повинні бути приємними для широкої аудиторії та відповідати критеріям вікового рейтингу 6+;
- на екрані завжди повинен рухатись якийсь графічний елемент. Це потрібно, щоб гравець не думав, що гра зупинилась.

Висновки до розділу 2

В цьому розділі були визначені вимоги до проєктування, розробки та кінцевого вигляду мобільного додатку. Ці вимоги достатньо описують функціонал та користувацький інтерфейс майбутнього додатку. Далі під час розробки, ці вимоги будуть спрямовувати рішення в правильному напрямку та допоможуть розробити цілісний продукт.

					IT61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		20

3 ТЕХНОЛОГІЇ ТА РІШЕННЯ ВИКОРИСТАНІ ПРИ РОЗРОБЦІ

Засобами розв'язання було обрано ігровий рушій Unity, об'єктно-орієнтоване програмування, мова C#, XML для збереження прогресу, мова програмування OpenGL для комп'ютерної графі C#, XML для збереження прогресу, мова програмування OpenGL для комп'ютерної графіки.

3.1 Ігровий рушій на Unity

Unity – це потужне середовище для розробки ігор для багатьох платформ та двигун виконання, що підтримує широкий спектр платформ. Використовуючи Unity. Розробники можуть розгорнути свої ігри на мобільних платформах (наприклад, Apple iOS, Google Android), консолях (Microsoft Xbox 360 і Xbox One, Sony PlayStation 3 і PlayStation 4 і Nintendo Wii, Wii U), портативні ігрові платформи (наприклад, Playstation Vita, Nintendo Switch), настільні комп'ютери (Microsoft Windows, Apple Macintosh і Linux), телевізійні скриньки (наприклад, Android TV і tvOS) та системи віртуальної реальності (VR) (наприклад, Oculus Rift, Steam VR, Gear VR). [1]

Основними цілями дизайну Unity є простота розробки та крос-платформи розгортання гри. Таким чином, Unity забезпечує простий у використанні інтегрований редактор середовище, в якому можна створювати та маніпулювати активами та організаціями які складають ваш ігровий світ і швидко переглядають вашу гру в дії там, у редакторі, або безпосередньо на вашому цільовому обладнанні.

Unity також забезпечує потужний набір інструментів для аналізу та оптимізації гри на кожній цільовій платформі, всебічний конвеєр кондиціонування активів та можливість однозначно керувати компромісом якості та якості на кожній платформі розгортання. Unity підтримує сценарії в JavaScript, C# або Boo. Має потужну анімаційну систему, яка підтримує ретаргетинг анімації (можливість відтворювати анімацію, створену для одного персонажа, на

					ІТ61.010БАК.004 ПЗ	Аркуш
						21
Зм.	Арк.	№ документа	Підпис	Дата		

абсолютно інший персонаж). [1]

Unity використовувались для створення найрізноманітніших опублікованих ігор, у тому числі Deus Ex: The Fall, Hollow Knight, Cuphead, короткометражний фільм “Адам” був знятий в режимі реального часу за допомогою Unity.

3.1.1 Розробка на Unity

3.1.1.1 Сцени, ігрові об’єкти та компоненти

В Unity геймплей створюється за допомогою сцен. Слід думати про кожен унікальний файл сцени як про унікальний рівень. У кожній сцені розміщуються свої ігрові об’єкти: оточення, перешкоди, прикраси. По суті гра проєктується та будується по частинам.

До ігрових об’єктів прикріплюються компоненти, які визначають його поведінку, вигляд та можуть модифікувати його будь-яким чином. [2]

Сцена, ігрові об’єкти та компоненти зображені на рисунку 3.1.

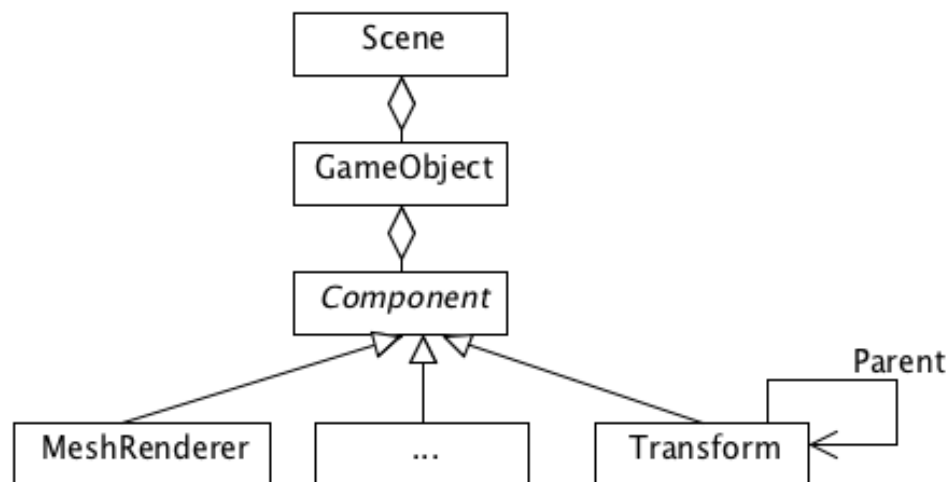


Рисунок 3.1 – Структура сцен в Unity

Така компонентна система надає ще декілька переваг ігровому рушію Unity. Проста та гнучка система прискорює створення гри та робить рушій легшим для освоєння.

3.1.1.2 Prefabs

Створення єдиної сцени та додавання до неї ігрових об'єктів може працювати для невеликих проектів, але в міру того, як проект збільшується в розмірах і складності, кількість ігрових об'єктів може швидко стати некерованими. Щоб вирішити це, Unity дозволяє створити з ігрового об'єкту Prefab.

Prefab – ігровий об'єкт збережений на диску.

Prefab система Unity дозволяє створювати, конфігурувати та зберігати ігрові об'єкти у комплекті з усіма їх компонентами, значеннями властивостей та дочірніми GameObjects, як багаторазовий об'єкт. Prefab об'єкт діє як шаблон, з якого ви можете створювати нові екземпляри збірних даних у сцені.

3.1.1.3 Створення екземплярів(instancing)

Створення екземплярів може бути корисним, коли потрібно створити багато копій одного і того ж об'єкта. Екземпляри можуть створюватись в редакторі та через код.

Всі екземпляри Prefab будуть змінені, якщо зміниться Prefab. Тому Unity надає функціонал для полегшення роботи з Prefab.

Prefab має функцію перекриття властивості. Тобто я можу створити екземпляр з Prefab рівня та перекрити поле назва. Тоді при зміні Prefab рівня, зміняться і всі екземпляри створені з нього, але не зміниться поле назва в тому екземплярі, в якому було перекрито це поле.

На рисунку 3.2 зображено створення екземпляру Prefab в ієрархії сцени.

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		23

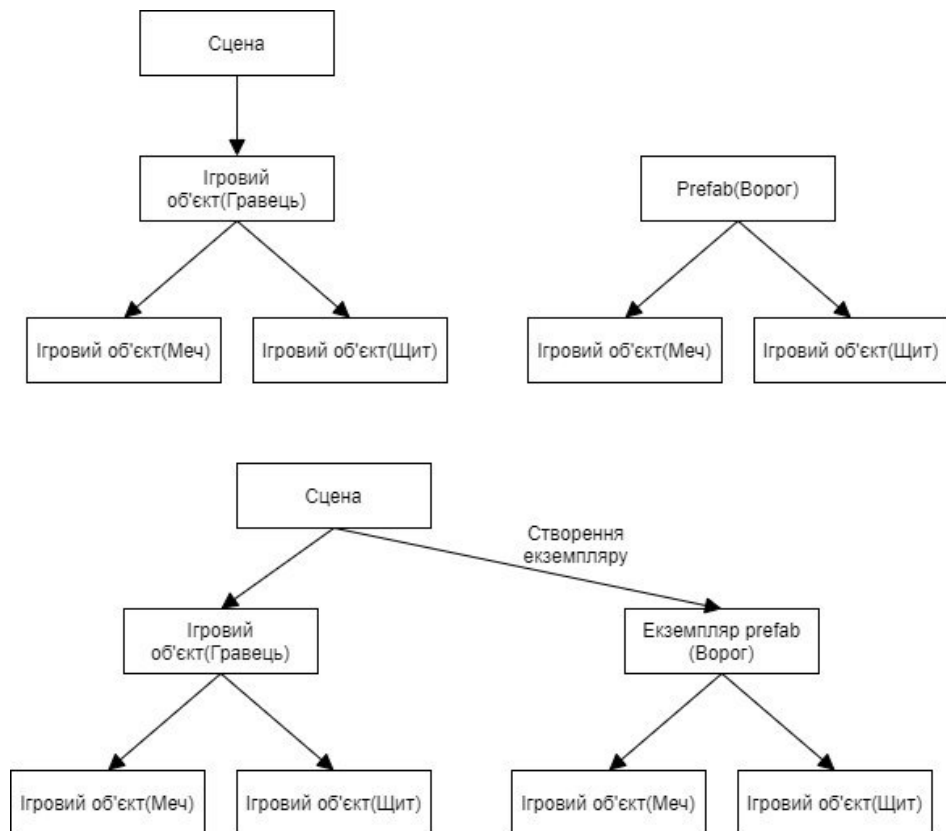


Рисунок 3.2 – Створення екземпляру Prefab

Таким чином можна досить легко створювати копії рівнів та не робити зміни для кожного рівня окремо, якщо вони потрібні всім.

3.1.2 Архітектура ігор в Unity

Створюючи ігри з Unity, рекомендований підхід полягає в тому, щоб почати, уявляючи видимі елементи у грі.

Наприклад, на рисунку 3.3 зображено архітектуру простої гри-шутер.

Після того як створена ієрархія об'єктів, як показано на рисунку 3.3, рекомендований процес створення гри – це створення prefab для кожного елемента, перерахованого на діаграмі. Потім будуть створюватись екземпляри (або через код, або безпосередньо в редакторі) для відносин власності. [3]

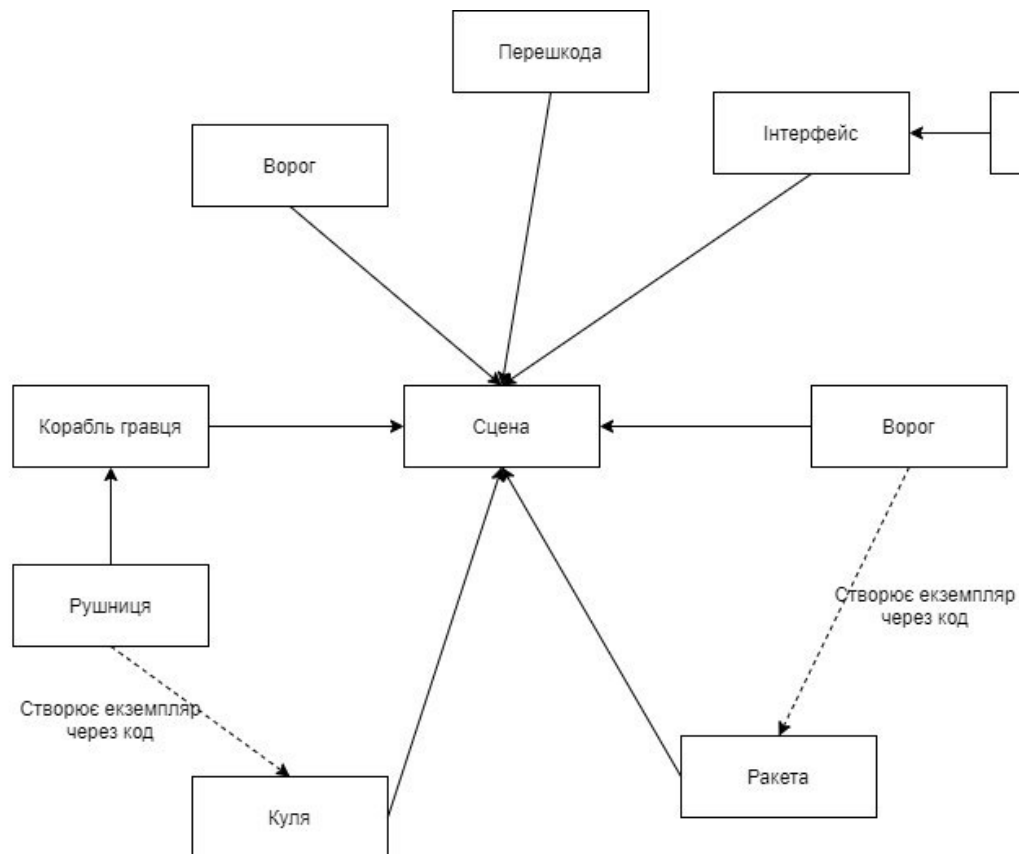


Рисунок 3.3 – Ієрархія об'єктів для гри-шутер

Проектування на основі сцен та prefab робить розробку набагато швидшою і простішою, що дозволяє зосередитись на самій логіці гри. Оскільки більшість ігрових компонентів відображаються безпосередньо на сцені, використання дизайну, заснованого на створенні екземплярів Prefab, означає, що мало іншого архітектурного коду потрібно.

3.1.3 Переваги Unity над іншими рушіями

Є багато інших комерційних ігрових двигунів. Більшість рушіїв розроблені в компаніях для власного користування. Але за останній час на ринку з'явилося достатньо якісних ігрових рушіїв таких, як Unity, Unreal Engine, GoDot, CryEngine. Всі вони безкоштовні.

Вибраним був Unity за рахунок ряду переваг над конкурентами:

- підтримка платформ. Двигун дуже вподобаний завдяки своїй розширеній підтримці до 27 платформ;

- інтегроване середовище розробки (IDE). Редактор розробок підтримує JavaScript та C# для створення сценаріїв, а також пропонує помітні функції, ідеальні для розробки ігор;
- графіка. Високоякісні аудіо та візуальні ефекти підтримуються двигуном, що полегшує розробку гри. Візуальні зображення пристосовуються на кожному екрані та пристрої без будь-яких спотворень чи компромісів з якістю зображення;
- документація. Unity має зрозумілу документацію;
- налагодження. Налагодження та налаштування надзвичайно простіше з розвитком гри Unity, оскільки всі ігрові змінні відображаються під час гри, що, в свою чергу, дозволяє розробникам налагоджувати процес під час виконання.

3.2 Мова програмування C#

C# – об'єктно-орієнтована мова програмування, розроблена Microsoft як частина платформи програмного забезпечення .net. Прагнення з моменту створення мови C# було створити просту, сучасну, об'єктно-орієнтовану мову з загальною метою. C# є основою для C++, Java і кілька мов, як Delphi, VB.net і C.

Перша версія C# була розроблена Microsoft між 1999 і 2002 і була офіційно введена в експлуатацію в 2002 в рамках платформи .net , яка спрямована на значне полегшення розробки програмного середовища Windows.

Однією з найбільших переваг .NET Framework є вбудована Автоматична система керування пам'яттю. Це звільняє програмістів від складного завдання управління пам'яттю для самих об'єктів і шукає правильний момент для його звільнення. Це значно підвищує продуктивність розробників і підвищує якість програм, написаних на C#. [4]

Спеціальний компонент CLR називається "збирач сміття" піклується про управління пам'яттю в .NET Framework. Основне завдання системи сміттевого збирача полягає в тому, щоб контролювати, коли пам'ять, відведена на змінні і об'єкти, більше не використовується, щоб звільнити його і зробити його

					IT61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		26

доступним для подальшої створення нових об'єктів. [5]

Зараз в ігровій індустрії домінуючою мовою програмування є C++.

C# має ряд наступних переваг над C++:

- програма компілюється на проміжну мову (CIL) незалежно від мови, на якій він був розроблений, або цільової архітектури та операційної системи;
- автоматичне вивезення сміття;
- швидша розробка програм;
- більш високий рівень абстракції;
- рефлексійні можливості;
- визначення класів та функцій можна проводити в будь-якому порядку;
- декларація функцій та класів не потрібна;
- класи можна визначити в межах класів.

C# 8.0 – це остання версія C#. Якщо ви подивитесь на історію мов C, C# розвивається швидше, ніж будь-які інші мови. Завдяки Microsoft та потужній підтримці громади. C# спочатку був розроблений для запису клієнтських програм Windows, але сьогодні C# може зробити майже все, що стосується консольних програм, хмарних додатків та сучасного програмного забезпечення машинного навчання.

3.2.2 C# та Unity

Як і майже всі сучасні двигуни, Unity використовує "мову скриптів", щоб ви могли змінити поведінку елементів гри. У контексті ігрового двигуна зазвичай йдеться про розміщену, інтерпретовану, доменну мову, більшість з яких виготовляються на замовлення, як GDScript Godot, GML GameMaker або GMS-програми Unreal.

Інші двигуни повторно використовують наявні розміщені мови, як Lua або JavaScript. Перш ніж Unity обмежив свою підтримку C#, він також підтримував JavaScript та Boo.

Використання мови C# має ряд переваг над розробленням та використанням

					IT61.010БАК.004 ПЗ	Аркуш
						27
Зм.	Арк.	№ документа	Підпис	Дата		

своїї скриптової мови.

Перша перевага, мова C# є більш універсальною ніж всі інші скриптові мови.

Друга, розробнику не потрібно вивчати нову мову, щоб використовувати двигун.

Але є й недоліки. Скриптова мова набагато легша за C# та ідеально підходить для дизайнерів. Також компонентами, які написані на C++ може керувати скриптова мова і таким чином забезпечується продуктивність C++ та легкість скриптових мов.

3.3 Open Graphics Library

OpenGL (Open Graphics Library) є потужним API для реалізації легко портативних графічних додатків. OpenGL є одним з найпопулярніших інтерфейсів програмування для реалізації 2D і 3D графіки. До цього сприяють широка доступність, сумісність з різними операційними системами та різними комп'ютерними платформами. Ця бібліотека підходить для додатків, що вимагають високої якості зображення в поєднанні з хорошою продуктивністю, щоб зробити його можливим, щоб генерувати його в режимі реального часу. [6]

OpenGL має низку переваг над іншими графічними API:

- OpenGL працює на багатьох платформах (Windows, Linux, Mac, Android);
- OpenGL має більш детальну документацію;
- OpenGL має менші накладні витрати на процесор, ніж Direct3D.

Основна функціональність Graphics Pipeline полягає в перетворенні вашої 3D сцени з урахуванням певного положення камери та орієнтації камери на 2D зображення, що представляють тривимірну сцену з точки зору цієї камери. Почнемо з огляду цього графічного конвеєра для схеми візуалізації на основі трикутника у наступному параграфі. У наступних параграфах буде розроблено конкретні компоненти. [7], [8]

На рисунку 3.4 зображено Graphics Pipeline, який працює на GPU. Це буде

					IT61.010БАК.004 ПЗ	Аркуш
						28
Зм.	Арк.	№ документа	Підпис	Дата		

використовувати наша програма, що працює на процесорі та являється вихідною точкою для графічного конвеєра. Програма буде нести відповідальність за створення вершин, і вона буде використовувати 3D API, щоб доручити процесору малювати ці вершини на екрані. [9]

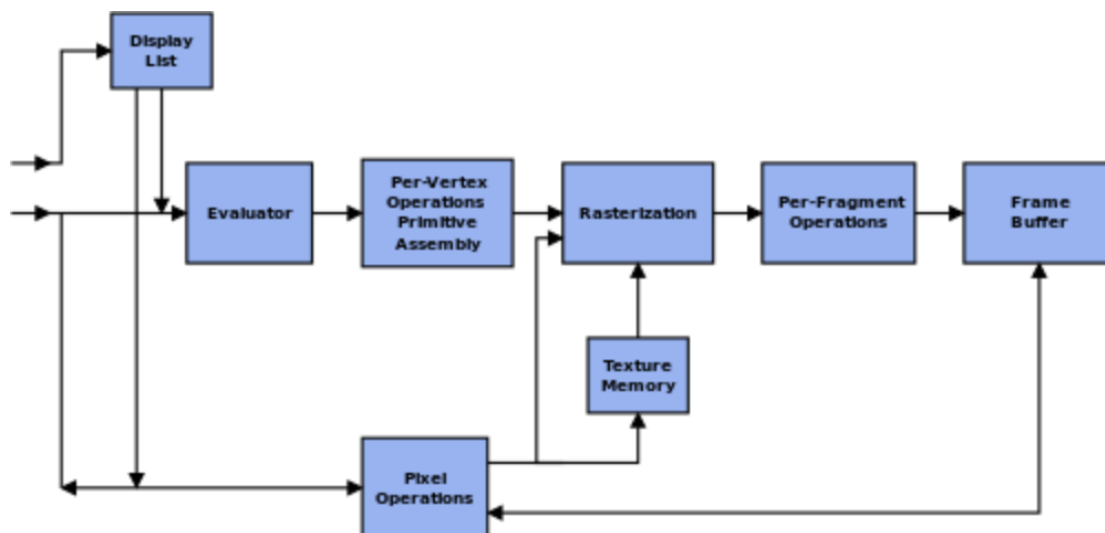


Рисунок 3.4 – Graphics Pipeline

Етапи Graphics Pipeline, які зображені на рисунку 3.4:

- Vertex Processing . Кожна вершина, отримана з вершинних масивів (як визначено VAO), діє за допомогою вершинного шейдера. Кожна вершина в потоці обробляється по черзі у вихідну вершину;
- Vertex Post-Processing, результати останнього етапу коригуються або відправляються в різні місця;
- примітивні відсікання, розділення точки зору та перегляд області перетворення у вікно;
- перетворення сканування та інтерполяція примітивних параметрів, що створює ряд фрагментів;
- Fragment шейдер обробляє кожен фрагмент. Кожен фрагмент генерує ряд результатів;
- Per-Sample Processing, застосовується для різних тестів.

В Unity є можливість отримати доступ до Graphics Pipeline та написати свій шейдер, який зможе виконуватися на стадіях Vertex Processing, Primitive Assembly

та Per-Fragment Operations та відповідно до цих стадій можна написати свої шейдери: Vertex шейдер, Geometry шейдер та Fragment шейдер. Вони будуть написані на ShaderLab, мова яка розроблена Unity. [10]

3.4 TexturePacker

TexturePacker – це утиліта для графічного дизайну, яка допомагає художникам текстур впорядкувати процес створення спрайтових аркушів, з автоматизованою підтримкою зображень високої роздільної здатності, масштабування та публікації готових проєктів на всіх сумісних пристроях в один клік. Додаток підтримує імпорт усіх основних форматів зображень (PSD, PNG, TGA, JPG, TIFF, BMP та SWF), імпортування файлів чи папок, імпортування файлів чи папок, автоматичного збирання та організації спрайтів (із збереженою структурою папок у складі імена спрайтів).

TexturePacker здатний упакувати декілька аркушів спрайту одночасно, створити додаткові аркуші, щоб створити місце для всіх спрайтів, сортувати спрайт по конкретних категоріях. Використовуючи Texture Packer Tool, візуальний, інтерактивний або ігровий проєкт зменшує кількість файлів, які він потребує для завантаження, зменшивши навантаження на клієнтські системи, економляться ресурси пам'яті, збільшить частоту кадрів, зробить анімацію плавною, збільшить час завантаження ігор та збережить візуальна якість навіть після того, як всі спрайти проходять через ефективне стиснення.

Додаток також може обрізати та обрізати ваші спрайти, таким чином зменшивши їх розмір текстури, зменшивши розмір аркушів спрайтів та забезпечивши швидше візуалізацію двовимірного мистецтва.

Щоб забезпечити більш зручне використання для розробників ігор, він має вбудовану підтримку для всіх основних багатопланових двигунів, орієнтованих на 2D, які зараз популярні. Сюди входять Unity, HTML5 / CSS Sprites, Cocos2D, SpriteKit, Starling, Sparrow, LibGDX та багато інших.

					ІТ61.010БАК.004 ПЗ	Аркуш
						30
Зм.	Арк.	№ документа	Підпис	Дата		

Висновки до розділу 3

У даному розділі були розглянуті всі технології, що будуть використовуватись при розробці даного проєкту.

Всі розглянуті технології є безкоштовними. Та мають велику популярність, що допоможе знайти навчальний матеріал.

Обрані технології дозволяють швидко реалізувати проєкт та дозволять підтримувати його.

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		31

4 РЕАЛІЗАЦІЯ ДОДАТКУ

У цьому розділі буде спроектовано та реалізовано інтерфейс, основні механіки гри та графіка. Також буде розглянута їх реалізація.

4.1 Реалізація рівня

Першим етапом розробки проєкту буде розробка рівню, що представляє собою вправу на набір тексту.

При розробці рівню були реалізовані три головні механіки:

Перешкоди з текстом над ними. Перед гравцем будуть з'являтися перешкоди, та він повинен знищити їх шляхом набору тексту, який знаходиться над ними.

Перешкоди поступово з'являються та рухаються в напрямку гравця. Якщо гравець не встигає набрати текст над перешкодою до зіткнення з нею, тоді рівень закінчується.

Точність набору гравця. Вона рахується від кількості неправильно та правильно набраних літер. Та гравець проходить рівень тільки, якщо його точність після завершення рівня становить більше 80%.

Так гравець повинен швидко та з високою точністю набирати слова, які з'являються перед ним.

Гру можна описати наступним списком правил:

- гравець вводить літеру. Якщо вона є першою літерою наступної перешкоди, тоді кількість літер над перешкодою зменшується та збільшується точність гравця. В іншому випадку точність гравця зменшується;
- гравець продовжує набирати літери, поки не закінчатся літери над перешкодою. Якщо це була остання перешкода, тоді гравець виграє. В іншому випадку з'являється нова перешкода, з новим набором літер;
- якщо гравець досягне перешкоди, коли він ще не встиг набрати всі літери, тоді гравець програє;

					ІТ61.010БАК.004 ПЗ	Аркуш
						32
Зм.	Арк.	№ документа	Підпис	Дата		

- якщо гравець виграв рівень та результат був кращим за попередній, тоді цей результат зберігається;
- якщо гравець в перший раз пройшов рівень, тоді в головному меню для гравця відкриється наступний рівень.

Для реалізації даного функціоналу було створено ієрархію об'єктів, як показано на рисунку 4.1. Данна ієрархія чітко розділяє гравця, всі елементи світу та інтерфейс.

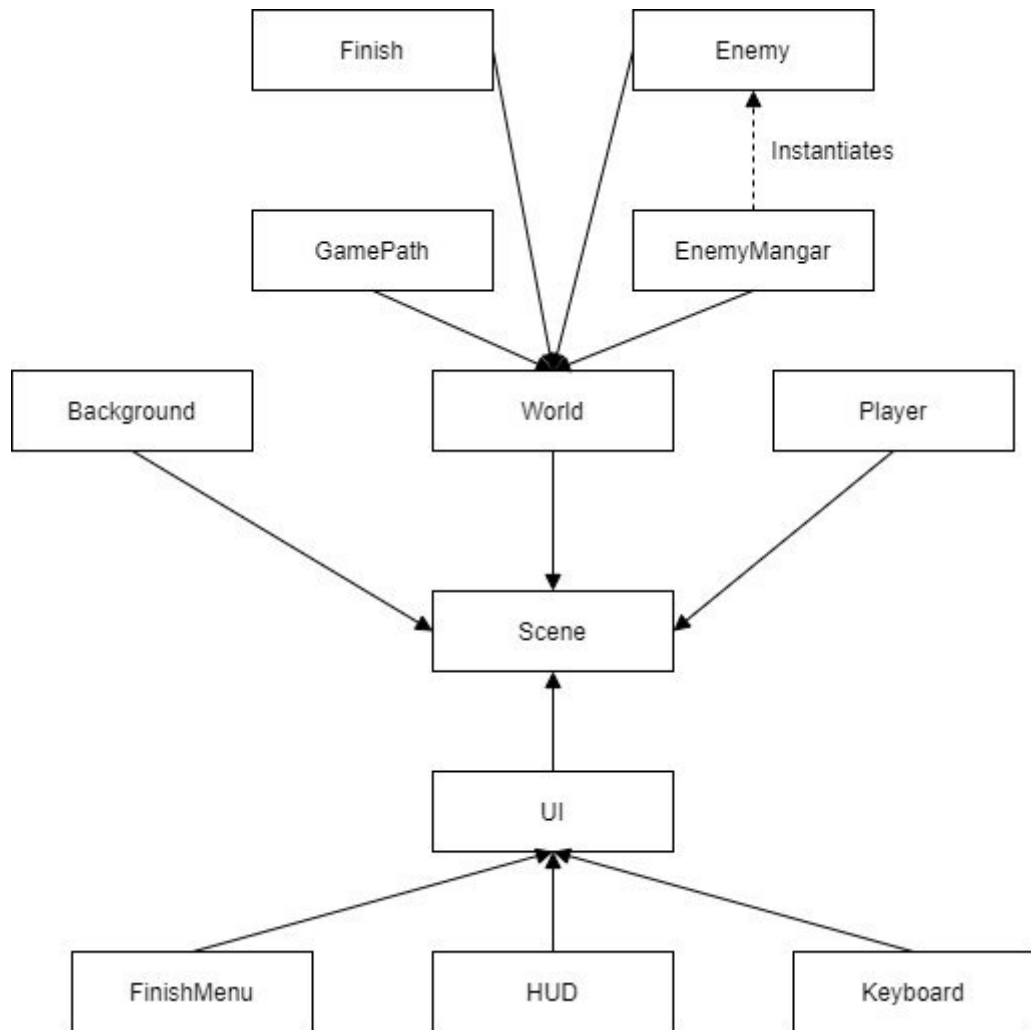


Рисунок 4.1 – Ієрархія об'єктів в сцені

З рисунку 4.1 видно, що в нас три основні об'єкти:

- світ (World). Він рухається за гравця, містить шлях його пересування (GamePath), фініш (Finish), та менеджер перешкод (EnemyManager), який створює перешкоди (Enemy);

- гравець (Player);
- інтерфейс (UI). Він складається з меню фінішу (FinishMenu), клавіатури (Keyboard) та HUD (Head-Up Display).

Другим кроком було створено системи, які далі були додані до об'єктів в сцені як компоненти. Реалізовано було наступний ряд систем:

- GameManager, ігровий менеджер – відповідає за стан рівню, за збереження результатів гри. Є сполучною ланкою між окремими елементами архітектури гри;
- InterfaceManager, менеджер інтерфейсу – відповідає за інтерфейс користувача, включаючи графічний інтерфейс;
- KeyboardManager, менеджер клавіатури – відповідає за ввід символів з різних клавіатур. В грі підтримуються дві клавіатури PC та Android. В класі KeyboardManager визначено поле IKeyboard, яке містить реалізацію клавіатури на даній платформі;
- AccuracyCounter – відповідає за точність набору. Цей клас підписаний на event набору вводу правильної та неправильно відповіді, тому він зберігає кількість відповідей та які з них були не вірними, потім оновлює значення точність гравця в цьому рівні;
- UnitManager, менеджер перешкод – відповідає за перешкоди. Він їх створює, знищує, та перевіряє чи настигли вони гравця.

Також UnitManager використовує вбудований компонент Unity – колайдер. Компонент колайдер визначає форму GameObject для фізичних зіткнень. За його допомогою перевіряється зіткнення перешкод з гравцем та визначаємо момент коли гравець дійшов до фінішу.

Всі менеджери реалізовані на основі патерну Singleton (одинак). Вони мають тільки один екземпляр і є загальними для всієї ігрової системи. Звернення до них відбувається за типом. [11]

Основні ігрові об'єкти, такі як перешкоди та гравець, реалізовані на основі патерну Finite State Machine (кінцевий автомат). Це дозволяє легко контролювати стан ігрового об'єкта і управляти його поведінкою. [11]

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		34

Реалізацію рівня можна побачити на рисунку 4.2.

4.2 Реалізація Head-Up Display

HUD (Head-Up Display) – це метод, за допомогою якого візуально передається інформація гравцеві як частина користувацького інтерфейсу гри. HUD часто використовується для одночасного відображення декількох відомостей, включаючи стан здоров'я головного героя, предмети та вказівки прогресування гри (наприклад, оцінка чи рівень).

В грі були реалізовані наступні елементи HUD:

- назва рівню, який зараз запущений. Він відображається зверху в центральній частині екрану;
- точність набору гравця. Відображається як процент правильних відповідей та знаходиться в правому верхньому кутку.

Текст, який відображається над перешкодами, змінює свій колір в залежності від вводу літер гравцем. Всі літери, які вже були набрані відображаються світло синім кольором, ще не набрані – білим. Якщо гравець набирає не вірну літеру, то літера, яку потрібно було ввести, один раз блимає червоним кольором.

Всі елементи HUD демонструють інформацію, яка потрібна гравцю в момент гри. Важливим є, що інформацію постає в зрозумілому виді та не заважає ігровому процесу.

Для реалізації тексту над перешкодами, використовувався окремий інструмент TextMesh Pro. Він має ряд переваг над звичайним текстом:

- TextMesh більш легкий, йому потрібно менше компонентів;
- TextMesh використовується у світовому просторі і вимагає певної математики, щоб він був у просторі екрану.

Реалізовані елементи HUD можна побачити на рисунку 4.2.

					ІТ61.010БАК.004 ПЗ	Аркуш
						35
Зм.	Арк.	№ документа	Підпис	Дата		



Рисунок 4.2 – Перший рівень

4.3 Анімації

Використані в цій грі анімації можна поділити на три типи:

- лінійний рух. Так в грі гравець стоїть на місці та весь світ рухається на гравця. Цей рух є лінійним та дає ілюзію, що гравець рухається вперед. Також симулюється ефект паралаксу, гори на задньому фоні повільніше за платформу на якій стоїть гравець;
- стандартні анімації інтерфейсу. Натиск кнопки, відкриття меню. Ці анімації вже були реалізовані в Unity. В проєкті лише було налаштовано їх зовнішній вигляд;
- використання систем Animation Clip та Animation Controller для анімації гравця.

4.3.1 Система Animation в Unity

Система Animation складається з двох компонентів Animation Clip та

					IT61.010БАК.004 ПЗ	Аркуш
						36
Зм.	Арк.	№ документа	Підпис	Дата		

Animator Controller. [12]

Animation Clip містить інформацію про те, як певні об'єкти з часом повинні змінювати своє положення, обертання чи інші властивості. Кожен кліп можна розглядати як єдиний лінійний запис. Анімаційні кліпи із зовнішніх джерел створюються художниками чи аніматорами із сторонніми інструментами.

Систему можна побачити на рисунку 4.3. Там зображений Animation Clip для анімації бігу гравця. Кожна точка на рисунку це зміна положення частини гравця в певний час.

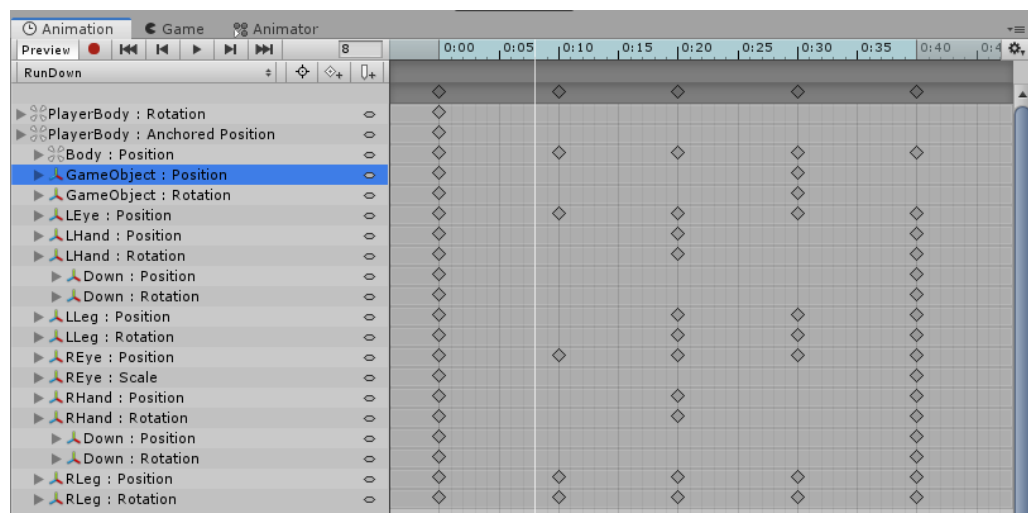


Рисунок 4.3 – Animation Clip

Потім анімаційні кліпи організовуються в структуровану систему, схожу на блок-схему, що називається контролер аніматора. Animation Controller діє як Finite State Machine («кінцевий автомат»). Який відслідковує, який саме в даний момент кліп слід відтворювати, і коли анімації повинні змінюватися або поєднуватися разом. [13]

Дуже простий Animator Controller може містити лише один або два Animation Clip, наприклад анімувати відкривання та закривання дверей у потрібний час. Більш просунутий контролер Animator може містити десятки гуманоїдних анімацій для всіх дій головного героя і може поєднуватися між декількома Animation Clip одночасно, щоб забезпечити плавний рух, коли гравець рухається по сцені.

Так можна побачити реалізацію анімацій для гравця на рисунку 4.4. Анімація має 5 станів:

- спокій(Idle);
- біг(Run);
- біг вгору(RunUp);
- біг вниз(RunDown);
- смерть(Die).

Також можна параметри анімації на рисунку 4.4. Ці параметри потрібні для встановлення умов для переходу між станами. Так, як показано на рисунку 4.4, якщо гравець буде в стані Run, RunUp або RunDown та в цей момент активується тригер die, тоді Animator Controller перейде в стан Die та відіграється анімація смерті.

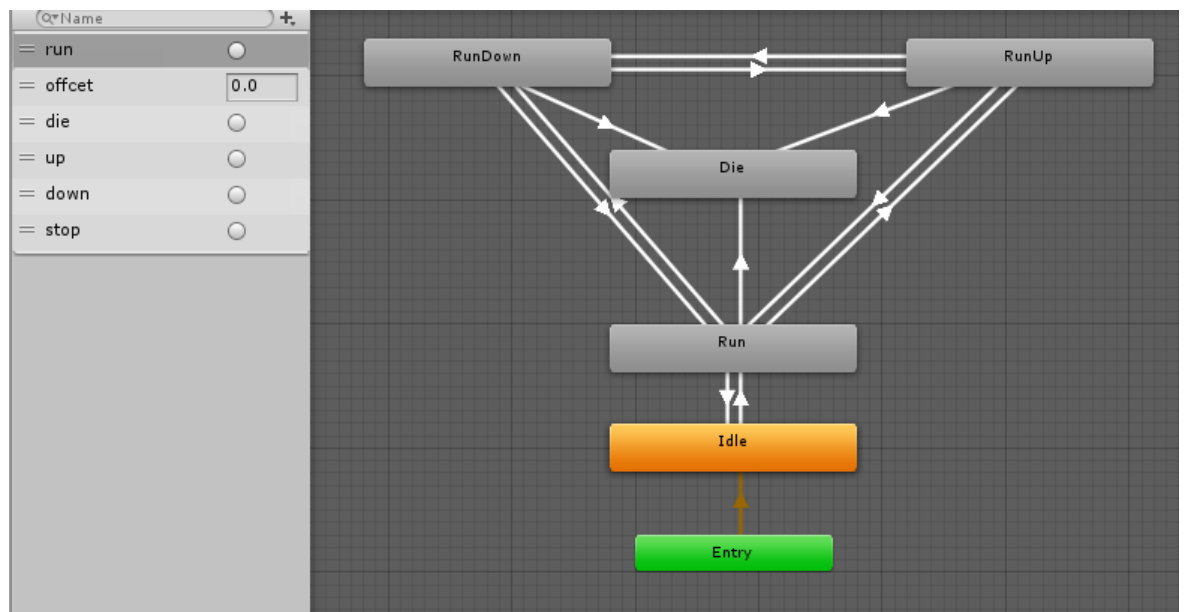


Рисунок 4.4 – Animator Controller

Для кожного з цих станів був створений окремий Animation Clip. Який програватиметься коли Animator Controller в стан з цим Animation Clip.

Для реалізації переходу станів в Animator Controller, в проєкті були реалізовані класи PlayerAnimTriggers та EnemyAnimTriggers. Вони зчитують стан гри, або стан перешкоди та в відповідь на зміну стану, відіграють потрібну

анімацію.

Для анімації гравця та перешкоди були використаний ще один компонент – Sprite Mask. Sprite Mask використовуються для приховування, розкриття частин спрайта або групи спрайтів. Але він працює як шейдер, який використовує stencil test.

Stencil Test – це операція на зразок, яка виконується після Fragment Shader. Значення фрагмента тестується на значення в поточному буфері трафарету. Якщо випробування не вдалося, фрагмент викреслюється.

Цей компонент потрібен для створення затухання гравця, коли відіграє анімація смерті.

4.3.2 Анімація пересування по рівню

Була реалізована проста лінійна анімація руху світу з ефектом паралаксу на фоні, коли гравець біжить прямо. Але для анімації підйому на гору та спуску цього не достатньо. Тому був доданий клас Waypoint, який зберігав всі точки та клас WaypointMover, який відповідав за рух по точкам. Точки задаються через редактор Unity та їх можна побачити на рисунку 4.5, вони відмічені синім кольором та мають цифри 01, 02, 03. Це фактично являється шляхом нашого гравця. [14]

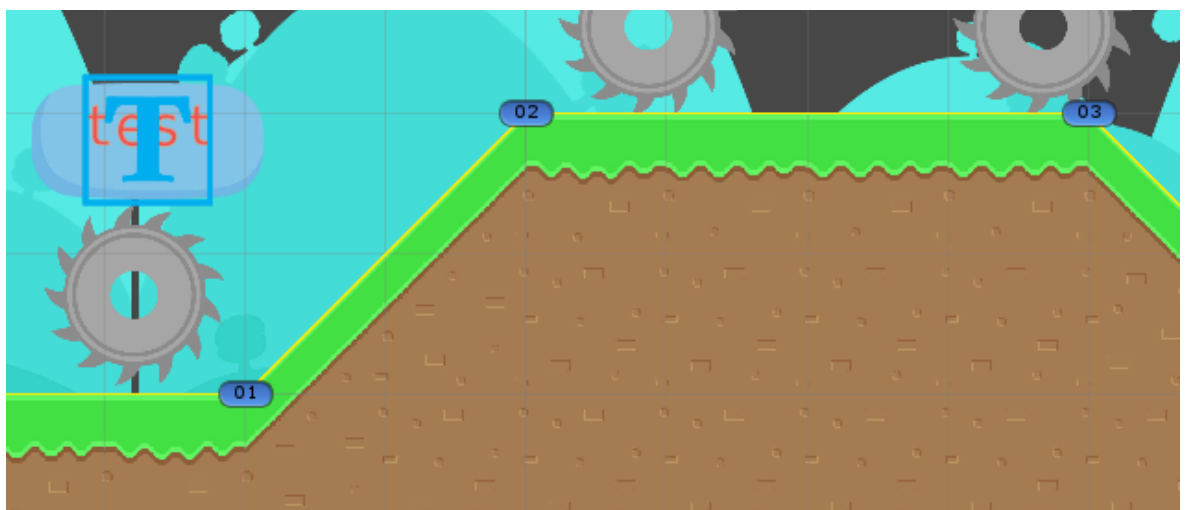


Рисунок 4.5 – Шлях гравця

					IT61.010БАК.004 ПЗ	Аркуш
						39
Зм.	Арк.	№ документа	Підпис	Дата		

Така система є досить гнучкою та допомогла вирішити ряд завдань:

- визначити в який момент гравцю потрібно спускатися або підніматися.

В цей момент повинна відіграватися відповідна анімація;

- отримати дані висоти підйому;
- допомогла реалізувати ефект паралаксу при зміні висоти;
- допомогла реалізувати зміну положення камери при спуску або підйому.

4.4 Графіка

Перш за все, варто виділити основні поняття, такі як тайл (Tile) і спрайт (Sprite).

Тайл (Tile) – двовимірний фрагмент, який повторюється та є будівельним блоком для побудови більших зображень (Тайлова графіка). Часто використовується в двовимірних іграх. [15]

Спрайт (Sprite) – графічний об'єкт, який представляє собою растрове зображення. Використовується в комп'ютерній графіці як основна одиниця для анімацій двовимірних об'єктів.

Більшість спрайтів були взяті з Unity Asset Store. Це магазин, що містить бібліотеку безкоштовних та комерційних продуктів, які створюють Unity Technologies, а також члени спільноти. [16]

Для пришвидшення Prefabs и програми на мобільному телефоні було вирішено застосовувати сторонню програму TexturePacker Він здатний упакувати декілька спрайтів одночасно в один атлас, як показано на рисунку 4.7. Також зібрані атласи будуть корисні при використанні Tile Palette.

					ІТ61.010БАК.004 ПЗ	Аркуш
						40
Зм.	Арк.	№ документа	Підпис	Дата		

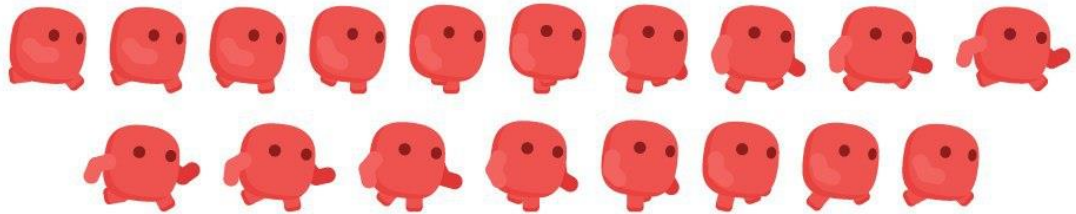


Рисунок 4.7 – Атлас анімації бігу гравця

Рівень можна зібрати прямо в вікні сцени Unity, але такий спосіб віднімати колосальну кількість часу, так як доводиться правильно позиціонувати тайли карти відносно один одного. З цієї причини, було вирішено використовувати функціонал Unity для реалізації даного завдання Tile Palette.

Tile Palette є вбудовану функцією Unity і надає широкий функціонал в плані реалізації двовимірних рівнів за допомогою тайлової графіки. Tile Palette використовується для малювання на Tilemaps. Tilemap вже змінюється в рівні та являє рисунок, нарисований тайловою графікою. На рисунку 4.6 зліва можна побачити Tilemap, справа – Tile Palette.

Цей інструмент був використаний для кожного рівня та головного меню. На рисунку 4.6 зліва видно вже готовий дизайн рівню, який був створений з тайлів вказаних в меню Tile Palette справа.

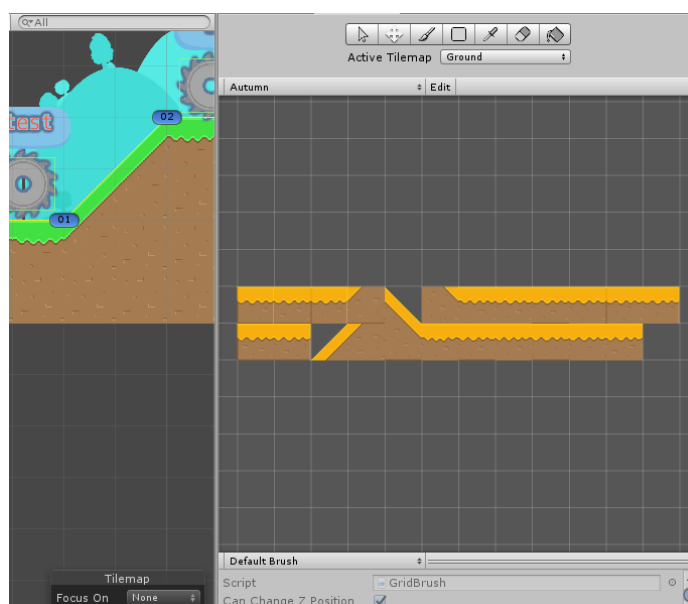


Рисунок 4.6 – Tile Palette

					IT61.010БАК.004 ПЗ	Аркуш
						41
Зм.	Арк.	№ документа	Підпис	Дата		

Tile Palette дає можливість створити декілька палітр зі спрайтів та використовувати їх на різних рівнях. Так у грі існує три варіанти одного спрайта платформи: весна, літо, зима. Тому були створені різні палітри з спрайтами та вони були використані на різних рівнях.

Такий інструмент значно пришвидшує створення ілюстрацій для нового рівня.

4.5 Інструменти для розробників

Для створення мобільного тренажера для друку потрібна велика кількість різноманітних рівнів. Також загострює проблему те, що ігри повинні випускати нові рівні навіть після виходу в магазин. Тому стоїть велика потреба в інструментах для створення нових рівнів.

В даному проєкті були реалізовані наступні інструменти для вирішення даної проблеми:

- гнучкий інструмент WaypointMover, розглянутий в пункті 4.2.2. Він нам дозволяє з легкістю змінювати маршрут гравця;
- Tile Palette, яка розглянута в пункті 4.3. Вона дозволяє нам мати декілька варіантів однотипних спрайтів та швидко створювати з них графіку для нових рівнів;
- налаштування рівня. Значна кількість параметрів за якими рівень може бути налаштований;
- система створення перешкод.

4.5.1 Налаштування рівня

Спочатку варто виділити поняття ScriptableObject.

ScriptableObject – це контейнер даних, який можна використовувати для збереження великої кількості даних, незалежно від екземплярів класу. Один з

					IT61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		42

головних випадків використання ScriptableObjects – зменшити використання пам'яті вашого проєкту, уникаючи копій значень. [17]

Створена система GameManager відповідає за стан рівню, за збереження результатів гри. Є сполучною ланкою між окремими елементами архітектури гри. Також вона приймає файл ScriptableObject. В цьому файлі містяться всі налаштування рівня.

Список налаштувань:

- текст, який повинен набрати гравець;
- швидкість руху гравця;
- швидкість появи нових перешкод;
- мінімальна швидкість появи нових перешкод. Якщо гравець не встигає за темпом рівня, то швидкість зменшиться до цього значення;
- максимальна швидкість появи нових перешкод. Якщо гравець не встигає за темпом рівня, то швидкість збільшиться до цього значення.

Кожен рівень має такий файл, тому змінивши всього декілька параметрів можна отримати зовсім інший рівень.

4.5.2 Система створення перешкод

Дана система повинна повністю залежати від налаштувань рівню. Перешкоди не повинні ставитись вручну.

Так був реалізований наступний функціонал:

- був створений об'єкт та клас Enemy. Як тільки він з'являвся в полі зору гравця, одразу ж об'єкт слово з черги слів, які потрібно ввести в цьому рівні. Потім він формує чергу Enemy, які повинен знищити гравець;
- перешкоди створюються за полем зору гравця з періодичністю, яка вказана в Scriptable Object до створеного для цього рівня;
- якщо в полі зору гравця більше трьох перешкод, тоді сповільнюється швидкість створення нових перешкод до значення, яке вказане в Scriptable Object;
- якщо в полі зору гравця одна перешкода, тоді пришвидшується

					IT61.010БАК.004 ПЗ	Аркуш
						43
Зм.	Арк.	№ документа	Підпис	Дата		

створення нових перешкод до значення, яке вказане в Scriptable Object.

Створена система дає нам можливість змінювати налаштування рівня, не змінюючи сцену.

4.6 Інтерфейс

4.6.1 Проєктування інтерфейсу

Для розуміння функціоналу, який буде потрібно буде реалізувати, було створено діаграму послідовності, її можна побачити на рисунку 4.7.

З рисунку 4.7 видно, що для реалізації інтерфейсу повинні бути створені два меню.

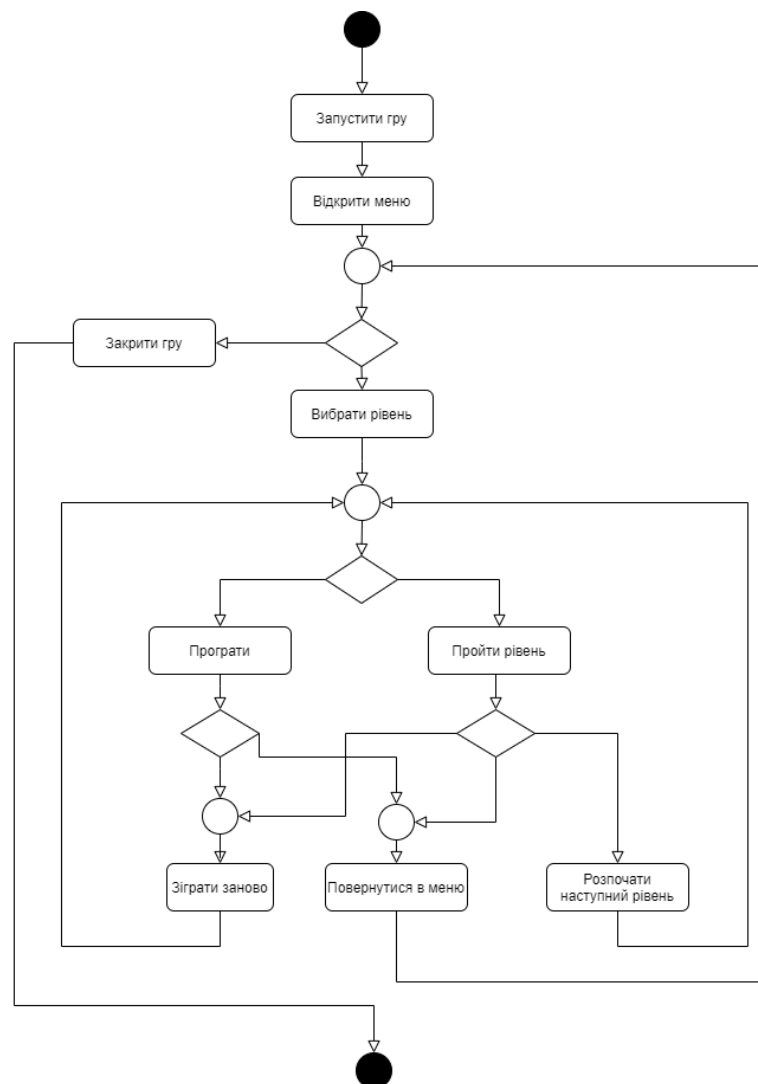


Рисунок 4.7 – Діаграма послідовності дій гравця

Перше меню, гравець побачить при вході в гру та воно міститиме вибір рівня або вихід з гри.

Друге, вже використовується в рівні, з'являється після завершення рівня та повинно містити такий функціонал:

- зіграти заново;
- повернутися в меню;
- розпочати наступний рівень.

4.6.2 Реалізація інтерфейсу

Реалізується даний інтерфейс за допомогою інструмента Canvas. Canvas – це область, на якій працює весь інтерфейс користувача елементи повинні бути всередині. Він відповідає за відображення елементів графічного інтерфейсу.

Також великою перевагою Canvas є можливість настроїти інтерфейс під різні розміри екрану. Це особливо важливо для мобільних девайсів, адже їх екрани можуть суттєво різнитися між різними моделями телефонів.

Після запуску додатку одразу на екрані буде відображена головне меню, це можна побачити на рисунку 4.8. Воно використовує різні TileMap для кожного рівня, щоб показати в якій палітрі виконаний рівень.

Головне меню має наступний функціонал:

- двері це кнопки які запускають різні рівні;
- недоступні рівні відображаються с замком та не спрацьовують після дотику;
- над пройденим рівнем відображаються точність набору, з якою пройшов його гравець;
- нові рівні відкриваються, якщо попередній був пройдений з точність більше 80%.

					IT61.010БАК.004 ПЗ	Аркуш
						45
Зм.	Арк.	№ документа	Підпис	Дата		



Рисунок 4.8 – Головне меню

Гравець вибирає рівень та натискає на нього. Перед ним з’являється рівень з елементом інтерфейсу “Tab to play”. Після його натиску гра розпочинається.

Після завершення рівня з’являється друге меню, як показано на рисунку 4.9. Воно різниться в залежності від результату з яким гравець закінчив рівень.

Меню має наступний функціонал:

- зіграти знову рівень;
- якщо у гравця відкритий наступний рівень або він пройшов цей рівень з точністю більше 80%, відкривається третя можливість – запустити наступний рівень;
- повернутися в головне меню.

					ІТ61.010БАК.004 ПЗ	Аркуш
						46
Зм.	Арк.	№ документа	Підпис	Дата		

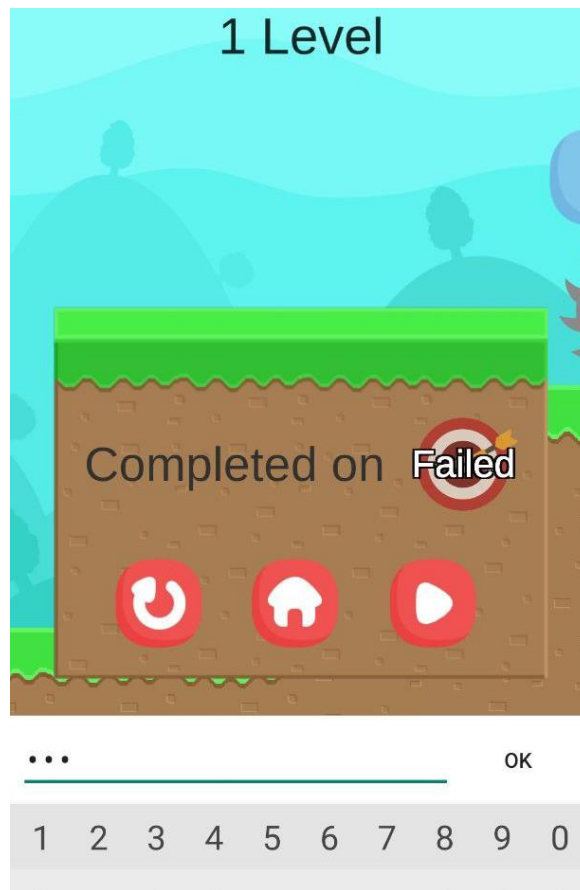


Рисунок 4.9 – Меню по закінченню рівня

4.7 Збереження прогресу користувача

Для реалізації цієї частини була використана система PlaterPrefs.

PlaterPrefs зберігає та надає доступ до налаштувань гравця між ігровими сеансами. Але в цьому контексті він був використаний для збереження прогресу.

Зберігалися два параметри:

- кількість пройдених рівнів;
- результати точності набору в кожному з них.

На Android дані зберігаються (зберігаються) на пристрої. Дані зберігаються в SharedPreferences. C# / JavaScript, Android Java та Native код можуть отримати доступ до даних PlayerPrefs. Дані PlayerPrefs фізично зберігаються в /data/data/pkg-name/shared_prefs/pkg-name.xml. [12]

					IT61.010БАК.004 ПЗ	Аркуш
						47
Зм.	Арк.	№ документа	Підпис	Дата		

Висновок до розділу 4

У даному розділі було переведено увагу до рішень проєктування та реалізації функціоналу та зовнішнього вигляду гри. Було спроектовано та реалізовано основні механіки, анімації, інструменти для розробників та інтерфейс.

Оптимізація графіки, наведені у цьому розділі, дозволила покращити швидкодію гри. Було спроектовано проєкт таким чином, щоб його можна було з легкістю розширити та досить просто підтримувати.

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		48

5 ТЕСТУВАННЯ

Оскільки тренажер було створено для мобільних телефонів з операційною системою Android, то для тестування було використано три пристрої з різними характеристиками.

Пристрої які використовувались для тестування:

- Xiaomi mi 8;
- Xiaomi mi 9 se;
- Redmi K20 Pro.

Технічні характеристики Xiaomi mi 8:

- розмір екрану 6.21 дюймів, 97.1 см²;
- розширення екрану 1080 x 2248 пікселів ;
- процесор Qualcomm SDM845 Snapdragon 845;
- внутрішня пам'ять 64GB 6GB RAM;
- GPU Adreno 630;
- операційна система Android 8.1.

Технічні характеристики Xiaomi mi 9 se:

- розмір екрану 5.97 дюймів, 87.5 см²;
- розширення екрану 1080 x 2340 пікселів ;
- процесор Qualcomm SDM712 Snapdragon 712;
- внутрішня пам'ять 64GB 6GB RAM;
- GPU Adreno 640;
- операційна система Android 9.0.

Технічні характеристики Redmi K20 Pro:

- розмір екрану 6.39 дюймів, 100.2 см²;
- розширення екрану 1080 x 2340 пікселів;
- процесор Qualcomm SM8150 Snapdragon 855+;
- внутрішня пам'ять 64GB 6GB RAM;
- GPU Adreno 640;
- операційна система Android 10.

					IT61.010БАК.004 ПЗ	Аркуш
						49
Зм.	Арк.	№ документа	Підпис	Дата		

Варто зазначити, що програма ще буде змінюватися, тому не має сенсу проводити велике тестування. Зараз в проєкті за мало рівнів та механік, щоб він зміг суттєво покращити навик швидкому друку. Саме з цієї причини дане тестування, є лише невеликою перевіркою на відповідність основним функціональним вимогам.

Тести охоплюють всі основні частини гри та було поділені на групи, які відповідають за окремий аспект гри. Список створених груп тестів:

- тести анімацій персонажу та перешкоди (Таблиця 5.1);
- тести анімацій (Таблиця 5.2);
- тести ігрових механік (Таблиця 5.3);
- додаткові тести (Таблиця 5.4);
- тест інтерфейсу на різних мобільних девайсах.

5.1 Тести анімацій персонажу та перешкоди

Анімації повинні бути починатися одразу після зміну стану гри. Також вони повинні плавно переходити з попереднього стану.

Всі стани анімацій були протестовані та результати тестування можна побачити в таблиці 5.1.

Таблиця 5.1 – Тести анімацій персонажу та перешкоди

Дія	Очікуваний результат	Підсумок
Почати гру	Персонаж починає бігти	Тест пройдений
Зачепити перешкоду	Програється анімація смерті персонажа та запускається анімація перешкоди.	Тест пройдений
Дійти до фінішу	Персонаж зупиняється	Тест пройдений
Персонаж добіг до підйому	Персонаж починає бігти вниз	Тест пройдений

Персонаж добіг до спуску	Персонаж починає бігти вгору	Тест пройдений
Написати слово	Програється анімація зникання перешкоди	Тест пройдений

Анімація, яка була створена для фону, камери та платформи відіграється при русі гравця. Вона була сильно відрізняється від анімацій гравця та перешкод та реалізована за допомогою інших інструментів тому вона була протестована окремо та результати тестування можна побачити в таблиці 5.2.

Таблиця 5.2 – Тести анімацій

Дія	Очікуваний результат	Підсумок
Почати гру	Платформа рухається на персонажа Фон рухається на персонажа повільніше за платформу	Тест пройдений
Зачепити перешкоду	Платформа та фон зупиняються	Тест пройдений
Дійти до фінішу	Платформа та фон зупиняються	Тест пройдений
Персонаж піднімається	Платформа та фон опускаються	Тест пройдений
Персонаж добіг спускається	Платформа та фон піднімаються	Тест пройдений

Основні ігрові механіки, які спрацьовують на рівні були протестовані, та результати тестування показані в таблиці 5.3.

Ігрові механіки часто змінюються, тому додаток повинен проходити тести вказані в таблиці 5.3 перед кожним оновленням. Також потрібно оновлювати сам список тестів з появленням нових механік.

Тести, що не відносились до жодної з груп тестів були згруповані в додаткові тести. Таблиця та результати тестування показані в таблиці 5.4.

Таблиця 5.3 – Тести ігрових механік

Дія	Очікуваний результат	Підсумок
Набрати коректну літеру	Літера, яку набрали повинна змінити колір на світло синій	Тест пройдений
Набрати неправильну літеру	Літера повинна блимнути червоним кольором	Тест пройдений
Набрати текст над перешкодою	Перешкода повинна зникнути, а текст залишитися	Тест пройдений
Зіткнутися з перешкодою	Гра повинна зупинитися	Тест пройдений

Таблиця 5.4 – Додаткові тести

Дія	Очікуваний результат	Підсумок
Пройти рівень та перезапустити гру	Результат пройденого рівня повинен зберегтись	Тест пройдений
Пройти рівень	В головному меню відкривається новий рівень	Тест пройдений

Також важливо протестувати адаптивність інтерфейсу та зручність керування на різних мобільних девайсах з різними розмірами екрану.

Гра тестувалася на 3 телефонах та при тестуванні головними були такі критерії:

- всі графічні елементи повинні залишатися в рамках екрану;
- елементи інтерфейсу не повинні перекривати одне одного;
- всі елементи інтерфейсу повинні функціонувати;
- кількість місця яке займає клавіатура, та як це впливає на ігровий процес.

Після тестування на трьох мобільних девайсах проблем з інтерфейсом не було виявлено.

Висновок до розділу 5

У даному розділі було проведення тестування всіх аспектів гри. Також був проведений тест інтересу на трьох мобільних девайсах. Всі тести пройшли успішно.

Дане тестування не гарантує відсутність проблем у кінцевого користувача, а лише зменшує вірогідність. Існує велика кількість девайсів з операційною системою Android, тому тести не можуть гарантувати коректність роботи програми на всіх девайсах.

					IT61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		53

6 ВИМОГИ ДО ТЕХНІЧНОГО ЗАСОБУ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА

6.1 Мінімальна конфігурація

Мінімальна конфігурація телефону:

- версія Android 4.4;
- процесор ARMv7 або ARM64;
- графічний API OpenGL ES 2.0+ або OpenGL ES 3.0+;
- оперативна пам'ять 1GB RAM.

6.2 Інструкція користувача

6.2.1 Установка програми

Для встановлення даного додатку потрібно буде:

- 15 Мбайт або більше вільного місця в пам'яті телефону;
- встановлений Google Play Store;
- доступ до інтернету.

Щоб встановити програму, достатньо відкрити Google Play Store та набрати в рядку пошуку “Typing Trainer”. Потім знайти додаток серед запропонованих результатів. Додаток в Play Store показано на рисунку 6.1.

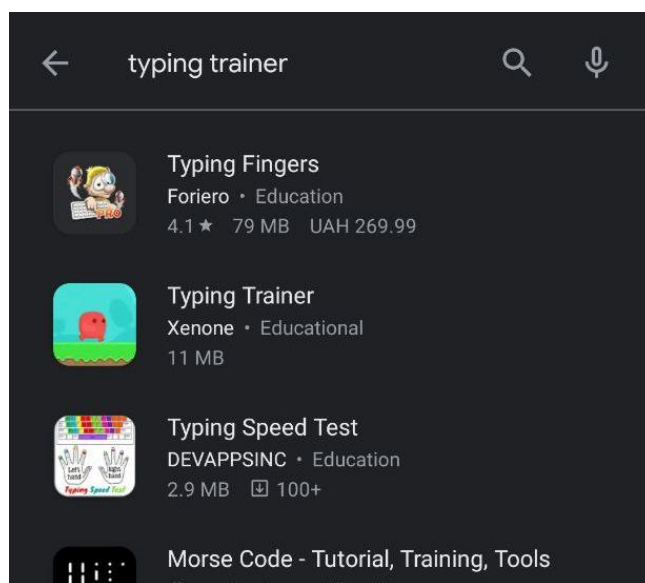


Рисунок 6.1 – Додаток в Google Play Store

					IT61.010БАК.004 ПЗ	Аркуш
						54
Зм.	Арк.	№ документа	Підпис	Дата		

Наступним кроком потрібно натиснути на додаток. Тоді з'явиться меню з рисунками з гри, коротким описом та кнопкою “Install”. Щоб встановити додаток, потрібно натиснути кнопку “Install”, тоді додаток встановиться та ви побачите ярлик на головному екрані телефону.

6.2.2 Використання програми

Після того, як програма була встановлена на вашому телефоні, ви можете запустити додаток з головного екрану в вашому телефоні. Після запуску ви потрапите в головне меню. Там будуть представлені всі доступні вам рівні та рівні які будуть відкриватися вам по проходженню гри.

Для запуску рівня потрібно натиснути на не заблоковані замком двері. Тоді завантажиться потрібний вам рівень, як показано на рисунку 6.2.

Щоб розпочати гру, потрібно натиснути на кнопку “Tap to play”, тоді замість кнопки з'явиться клавіатура, персонаж почне бігти та вам потрібно буде набирати літери, які написані над перешкодами.

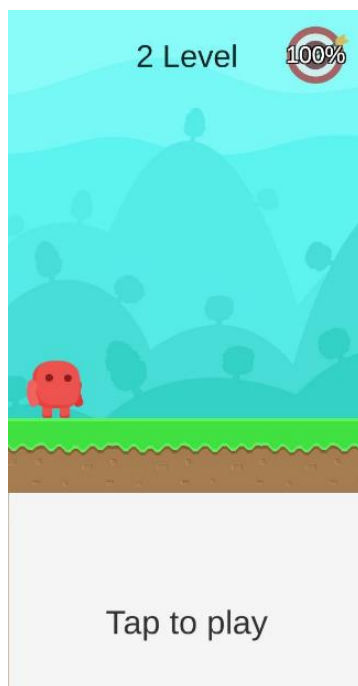


Рисунок 6.2 – Другий рівень

					IT61.010БАК.004 ПЗ	Аркуш
						55
Зм.	Арк.	№ документа	Підпис	Дата		

Після завершення рівню з’явиться меню з результатом. Тоді ви зможете або знову запустити цей рівень, або повернутися в головне меню, або почати наступний рівень. Кожен раз коли ви проходите новий рівень, відкривається один рівень в головному меню. Меню зображено на рисунку 6.3.

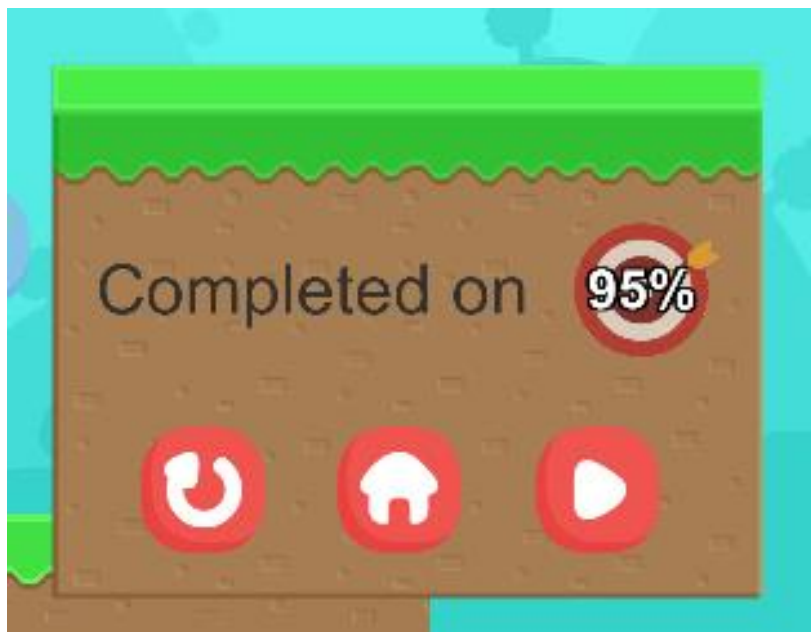


Рисунок 6.3 – Меню по закінченню рівня

Висновок до розділу 6

У даному розділі були встановлені мінімальні вимоги та інструкція для користувача. Цей розділ є дуже важливим для всіх нових користувачів даної системи.

					ІТ61.010БАК.004 ПЗ	Аркуш
						56
Зм.	Арк.	№ документа	Підпис	Дата		

ВИСНОВКИ

В даному дипломному проєкті було проаналізовано існуючі на ринку тренажери для друку. Були зроблені висновки та розроблені вимоги до проєкту, який би поєднував найкращі рішення на веб платформі та намагався би перенести їх на мобільні девайси. Для розробки проєкту проведений ретельний аналіз сучасних технологій, що дозволили реалізувати даний проєкт. Реалізовано архітектуру та обрані технології, які дозволяють швидко розширювати функціонал та підтримувати проєкт після його виходу. Далі були створені системи, що реалізовували всі потрібні нам механіки, давали змогу швидке додавання нових або зміну вже існуючих рівнів та відповідали за анімації і графічний інтерфейс. Також був розроблений інтерфейс, який відповідав всім вимогам інтерфейсу гри, він є зручний, приємний, швидко відповідає на вхідні сигнали та після кожного сигналу інформує гравця о зміні стану гри.

Створений додаток для мобільних телефонів призначений для тренування набору тексту. Він дозволяє швидко та в цікавій формі отримати або покращити навик друкування.

Розроблений додаток має ряд переваг на існуючими аналогами:

- додаток виконаний як гра, що робить процес набору тексту набагато цікавішим;
- приємна анімація та графіка. Використання сучасного рушія дозволило розробити анімації та графіку кращу ніж у існуючих аналогів;
- проєкт випущений на мобільну платформу, що робить його зручнішим в використанні;
- багатоплатформність. Даний додаток працює на великій кількості девайсів, та може бути випущений на інші платформи.

Для створення повноцінного продукту, який зміг би конкурувати з найкращими тренажерами на мобільній платформі, потрібно зробити наступні кроки:

- розширення кількості рівнів;

					IT61.010БАК.004 ПЗ	Аркуш
						57
Зм.	Арк.	№ документа	Підпис	Дата		

- додавання режиму гри на швидкість;
- реалізація можливості змагатися з іншими гравцями;
- розширення кількості мов;
- вдосконалення інтерфейсу користувача;
- реалізація нескінченного режиму гри.

					ІТ61.010БАК.004 ПЗ	Аркуш
Зм.	Арк.	№ документа	Підпис	Дата		58

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Game Engine Architecture. Third Edition” Джейсон Грегори 2020. 35 ст.
2. The component based scenegraph – Режим доступу:
<http://www.kickjs.org/tutorial/part1.html>
3. Офіційна документація рушія Godot – Режим доступу:
<https://docs.godotengine.org/en/stable/>
4. Рихтер Джеффри. CLR via C#. Програмування на платформі Microsoft.NET Framework 4.5 на мові C#. 4-е вид. / Джеффри Ріхтер.
5. C# in Depth, 3rd Edition Third Edition by Jon Skeet.
6. Сторінка з вправами та теорією по OpenGL - Режим доступу:
<https://open.gl/introduction>
7. “Rendering Hell 2.0” Саймон Шрайбт – Режим доступу:
<https://simonschreibt.de/gat/renderhell/>
8. Real-Time Rendering, Third Edition 3rd Edition by Tomas Akenine-Moller, Eric Haines, Naty Hoffman.
9. Сторінка вікі з документацією по OpenGL – Режим доступу:
https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview
10. OpenGL ES 3.0 Programming Guide (2nd Edition) by Dan Ginsburg, Budirijanto Purnomo, Dave Shreiner, Aaftab Munshi.
11. “Game programming patterns” Роберт Нистром – Режим доступу:
<https://gameprogrammingpatterns.com/>
12. Офіційна документація Unity – Режим доступу:
<https://docs.unity3d.com/Manual/index.html>
13. Refactoring guru [Електронний ресурс] – Режим доступу:
<https://refactoring.guru/ru/design-patterns>
14. Essential Mathematics for Games and Interactive Applications 3rd Edition 2016 by James M. Van Verth, Lars M. Bishop.
15. “Unity in action” Джозеф Хокинг – Режим доступу:
<https://www.manning.com/books/unity-in-action>

					IT61.010БАК.004 ПЗ	Аркуш
						59
Зм.	Арк.	№ документа	Підпис	Дата		

16. Unity Game Engine [Електронний ресурс]. - Режим доступу:
<https://unity3d.com/ru>

17. Building an FPS Game with Unity by John P. Do

					ІТ61.010БАК.004 ПЗ	Аркуш
						60
Зм.	Арк.	№ документа	Підпис	Дата		